mi computer

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

TOMO 1



mi COMPUTER

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

ISBN: 84-85822-84-6 (tomo 1) 84-85822-82-X (obra completa)
Depósito legal: B. 52/1984
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona)
Impreso en España - Printed in Spain - 1984

mi COMPUTER

VOLUMEN 1



¿Qué es un ordenador?

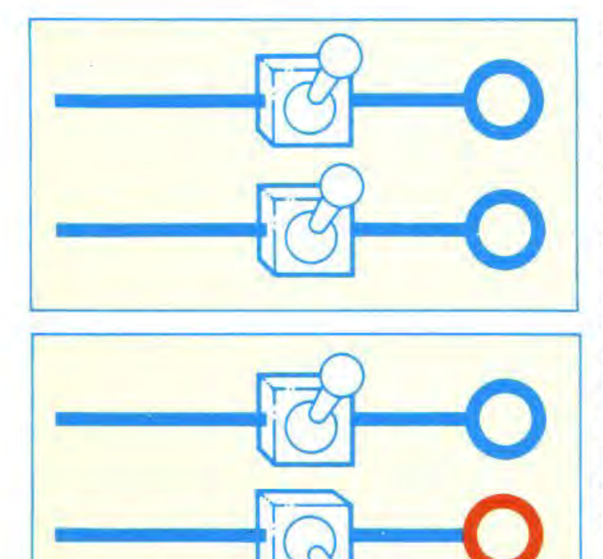
¿Cómo "piensan" los ordenadores y cuánto "saben"? Las respuestas son esenciales para llegar a entenderlos

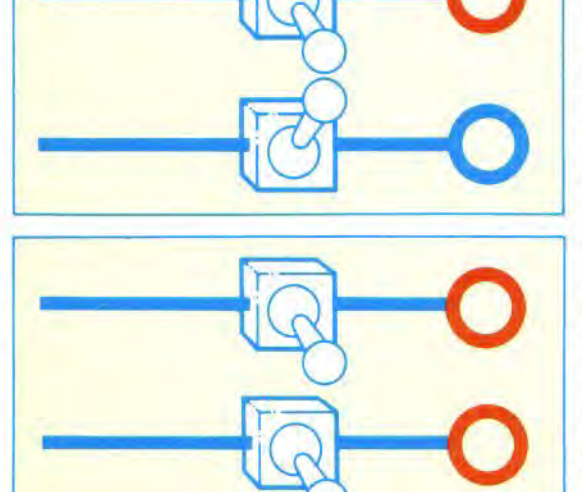


La pregunta "¿Qué es un ordenador?" no puede responderse con la misma sencillez que, por ejemplo, "¿Qué es un televisor?" o "¿Qué es una lavadora?". La razón de esta dificultad es que, a diferencia de estos dos electrodomésticos, el ordenador no posee una única función exclusiva. Los ordenadores digitales son una nueva generación de máquinas capaces de ejecutar una serie casi infinita de funciones según el programa que su propietario les suministre.

La idea de programabilidad, es decir, la capacidad de programación, no resulta del todo extraña en el hogar moderno; muchos electrodomésticos, como las lavadoras automáticas, llevan incorporados diversos programas que permiten utilizarlos de maneras diferentes. Sin embargo, en el caso de los ordenadores, es toda la función de la máquina la que, en cuestión de minutos, puede modificarse mediante la introducción de un nuevo programa: el ordenador se convierte en procesador de textos, en máquina para juegos o en un contable que cuida del estado de su cuenta bancaria.

¿Cómo realiza el ordenador tantos trabajos diferentes? Lo sabremos en profundidad a medida que avance el curso, pero analicemos ahora someramente algunos de los principios que entran en juego.







Conmutando en números
Para representar los números, los ordenadores utilizan circuitos eléctricos. Los circuitos consisten básicamente en interruptores. Cada interruptor puede estar en una de dos posiciones: encendido o apagado ("on" u "off"). Con dos interruptores se consiguen cuatro combinaciones de "on" y "off". Los ordenadores emplean un sistema similar a éste para representar los números. "Off/off" es cero, "off/on" es uno, "on/off" es dos y "on/on" es tres. La combinación de grupos compuestos por más de dos elementos permite la representación de mayor cantidad de números. Los ordenadores pueden procesar cifras muy elevadas y realizar complicadas operaciones matemáticas con gran rapidez, valiéndose de miles de interruptores microscópicos

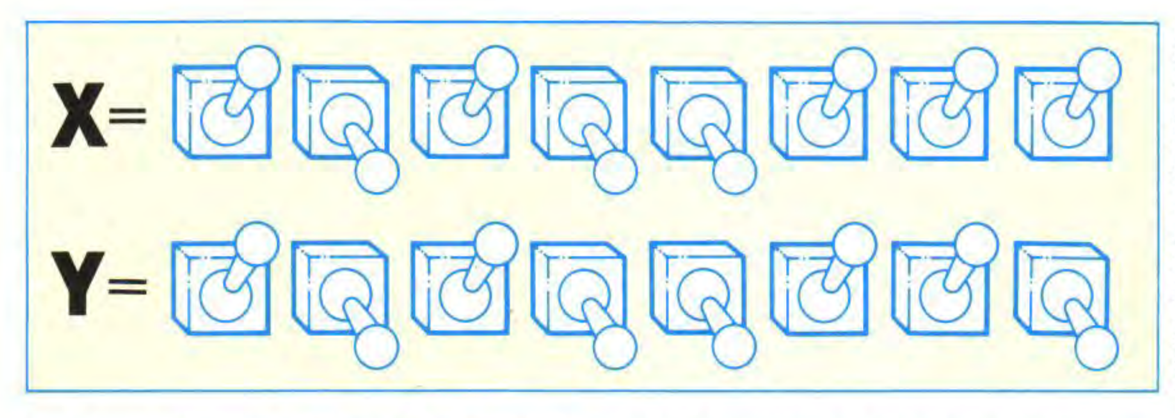
En cierto sentido, un ordenador no es más que una caja repleta de diminutos interruptores eléctricos que pueden conectarse entre sí de distintas maneras. Ésta no es, sin embargo, la mejor forma de comenzar si se desea comprender lo que un ordenador puede *hacer*; a este nivel no somos nosotros los interesados en comprender los ordenadores, sino quienes los diseñan y construyen. El ordenador moderno es una máquina extraordinariamente compleja; gracias al sorprendente desarrollo en el campo de la microelectrónica (el famoso *chip* de silicio, o circuito integrado de silicio), incluso hasta el más pequeño de los ordenadores personales puede contener 250 000 de estos pequeños interruptores. Todos estos interruptores pueden estar "encendidos" o "apagados". Cualquier apostador de quinielas sabe que el número de combinaciones posibles entre interruptores "encendidos" y "apagados" es enorme. Por otra parte, el ordenador que usted adquiera llevará un programa incorporado con carácter permanente que será como un disfraz para esta complejidad que resulta asombrosa para la mente de cualquier persona, y que permitirá que usted "hable" con la máquina mediante algunos vocablos abreviados pero fácilmente reconocibles.

Al utilizar un ordenador por primera vez, mucha gente se sorprende porque, una vez encendido, descubren que la máquina no sabe nada que les resulte útil. Por extraño que parezca, muchas personas siguen pensando, aún hoy, que el ordenador es un "cerebro electrónico" del cual se espera que lo sepa todo. ¿Es necesario que el ordenador sepa cuál es la capital de Afganistán? ¿O cuál es la altura del Kilimanjaro? En realidad, lejos de saber todas estas cosas, el chip de silicio, que conforma el "cerebro" de un ordenador, no sabe ni siquiera el alfabeto y menos aún algo de aritmética. Lo único que el ordenador sabe es varios centenares de combinaciones numéricas, y todo lo que se le quiera enseñar ha de ser indefectiblemente traducido a esos números. Los pequeños interruptores que hemos mencionado pueden recordar números; cada patrón de interruptores ON y OFF ("encendido" y "apagado", respectivamente) representa un número de acuerdo al sistema numérico binario, que sólo utiliza las cifras "0" y "1". El hecho de que el ordenador pueda recordar o, mejor dicho, *almacenar* información, es de vital importancia para la forma en que éste trabaja; la memoria electrónica de un Sinclair Spectrum retiene una información equivalente al contenido en palabras de seis páginas de este fascículo.

Además de almacenar números en su memoria, el ordenador puede hacer cosas *con* esos números: puede sumarlos y restarlos, compararlos entre sí y trasladarlos dentro de su memoria. Todo lo que la máquina puede hacer se deriva de estas sencillas operaciones. Supongamos que deseamos almacenar textos en el ordenador. Inventemos un código, de manera que a cada letra del alfabeto le corresponda un número determinado: el ordenador podrá entonces almacenar palabras en forma de números y combinarlas entre sí. ¿Que deseamos jugar a la rana? Dibujemos la silueta de una rana y tracemos sobre ella una cuadrícula, de modo que a cada pequeño cuadrado resultante pueda asignársele un número... Por supuesto, no es necesario que usted mismo se invente los códigos, pues este trabajo ya ha sido realizado por los fabricantes y los diseñadores, que lo han estructurado en forma de programas para ordenadores.

¿Qué es un programa? Es una lista de instrucciones que se le proporciona al ordenador para que realice operaciones simples (sumar, comparar, etc.) en un orden especial. Pero, ¿en qué consisten estas instrucciones y cómo llegan hasta el ordenador? ¡En realidad, se trata tan sólo de otros números que también están almacenados en la memoria del ordenador! Esto parece colocarnos ante una paradoja similar a la del huevo y la gallina. El ordenador no puede hacer nada si no cuenta con un programa que le diga lo que debe hacer; cada vez que usted pulsa la letra "A" en el teclado, un programa contenido en el ordenador debe explorar el teclado, descubrir cuál es la tecla que ha pulsado y luego decirle al ordenador cuál es el código numérico que le corresponde a esa letra. Pero en un principio, cuando se diseñaron los primeros ordenadores, este programa de exploración del teclado no existía; alguien debía colocar los números correctos directamente en la memoria del teclado, mediante instrumentos especiales, sólo para conseguir que ésta comprendiera las letras mecanografiadas y que estas letras quedaran reflejadas en una pantalla de televisión.

Pero una vez realizados estos primeros programas básicos, todo resulta mucho más sencillo. Ahora usted



Un código para letras y números

Un grupo de ocho interruptores permite 256 combinaciones distintas de "on" y "off". Esta cantidad resulta más que suficiente para un código individual (que no emplee más que unos y ceros) para cada una de las letras, números y signos especiales que se incluyen en el teclado de un ordenador similar al de una máquina de escribir. La ilustración muestra cómo están representadas las letras X e Y dentro de un ordenador que utiliza el código ASCII

puede introducir nuevos números en la memoria del ordenador mecanografiándolos en el teclado. Este proceso se denomina programación en código de lenguaje máquina, y más adelante nos ocuparemos de él con mayor amplitud. Como la programación en código de lenguaje máquina es bastante difícil y aburrida, algunos programadores ingeniosos han elaborado programas (en código de lenguaje máquina) que traducen los vocablos ingleses como PRINT, BEEP, LOAD y LIST (impresión, emisión de señal acústica, carga y listado de un programa) a instrucciones en código de lenguaje máquina que el ordenador puede utilizar. Todos los ordenadores personales, a excepción de los más sofisticados, llevan incorporado este tipo de programa, gracias a lo cual uno puede programarlos mediante un lenguaje sencillo para ordenadores en lugar de hacerlo mediante series de números. Este lenguaje se denomina BASIC. Recuerde que cada vez que utilice el BASIC lo que está trabajando para usted allí, en el interior del ordenador, es el producto de horas y horas de trabajo por parte de los programadores.

Con lenguajes de ordenador como el BASIC resulta sencillo escribir programas útiles o tan sólo de entretenimiento, mientras uno permanece ajeno a la frenética y compleja actividad que se ha suscitado en el interior del ordenador sólo para detectar que se ha tecleado la letra A. Por ejemplo, es muy fácil escribir un programa que almacene los nombres de las capitales de todos los países del mundo para que, al ser requerido el ordenador con la pregunta "¿Cuál es la capital de Afganistán?", proporcione como respuesta "Kabul". En otras palabras, el cerebro electrónico sólo puede saber aquello que se le ha comunicado con anterioridad: no puede descubrir cosas por sí mismo.

Si las cosas son así, ¿por qué, entonces, los ordenadores son tan útiles? Porque pueden almacenar enormes cantidades de información y manipularla mucho mejor que las personas. Y, por supuesto, no siempre tiene que ser *usted* quien coloque en él la información por primera vez. Puede comprar un programa, escrito por otra persona, que tenga almacenados los nombres de todas las capitales; en este caso el ordenador actuará como libro de referencia electrónico. Otra posibilidad es comprar un programa que trabaje con la información que usted le proporcione a través del teclado: un programa de "tratamiento de textos" con el que podrá mecanografiar, corregir y hacer nuevas copias de, por ejemplo, sus cartas y documentos, o bien un programa de "base de datos" con el cual podrá confeccionar el catálogo de una biblioteca y que en unos segundos le responderá a preguntas como "¿Qué libros de George Bernard Shaw poseo que hayan sido publicados antes de 1926?"

El hecho de que esa pobre máquina sorda que es el ordenador comprenda sólo números, en la práctica no es un signo de debilidad, sino más bien de fortaleza. Si los ordenadores trataran realmente con los objetos que nos interesan, pongamos por caso con palabras o con colores, serían muchísimo más complicados y se necesitaría una clase diferente de ordenador para cada tipo de trabajo que se realizara. De todas maneras, ¿cómo haría usted para almacenar VERDE en la memoria de un ordenador? Una vez se ha captado el principio de que el ordenador no necesita "comprender" aquello con lo que está tratando en la forma que una persona precisa comprenderlo, se entenderá cómo es que un solo tipo de ordenador puede ocuparse prácticamente de todo. Lo único que se requiere es que un programador sea capaz de describir el problema de

¿POR QUÉ SOFTWARE?



El micro es un maestro nato

Útil para cartas y estados de cuentas

Fue inventado con fines científicos

Y también para jugar y divertirse

Un ordenador es una máquina versátil y puede desempeñar diversas funciones. El software centraliza su poder. Una misma máquina puede ser utilizada por un hombre de negocios con un software de gestión, por un tecnólogo con un software estadístico, o con fines de entretenimiento si se carga con un software de juegos. El software es lo que determina la actividad del ordenador

Tony Lodge

Lo que sucede en el interior

Para instalar un sistema completo de ordenador y dejarlo a punto para ser utilizado, es necesario conectar entre sí diversas unidades. Los chips de silicio, responsables de la existencia del ordenador personal, están alojados en el interior de la caja, por lo general debajo del teclado. Quite la tapa y se encontrará con estos componentes principales

forma que éste pueda reducirse a una serie de números. Por ejemplo, si deseamos que un ordenador produzca música, no esperaremos hacer flotar en su interior los sonidos reales; en cambio, será necesario describir cada nota de la escala musical mediante un número que sea proporcional a su altura o su frecuencia. Podemos hacer que el ordenador envíe las señales eléctricas que utiliza para representar los números a un amplificador en lugar de a una pantalla de televisión, con lo cual se podrá oír los resultados. ¿Cómo hacer que un disparo de misil atraviese la pantalla dirigiéndose hacia los "invasores del espacio"? Simplemente, desplazando algunos números, que representan la forma de un misil, de un lugar a otro en la parte de la memoria del ordenador que actúa como "mapa" de la pantalla del televisor. Tanto a las imágenes como al movimiento, al color y al sonido puede asignárseles un código numérico adecuado que permita que el ordenador los manipule y que, mediante un "transmisor", como un televisor o un amplificador, lo vuelva a convertir en señales que tengan algún significado.

Intentaremos, pues, responder a la pregunta que nos formulamos al comienzo: "¿Qué es un ordenador?" Es una máquina que almacena señales electrónicas que representan números. Algunos de estos números son instrucciones que le dicen al ordenador lo que debe hacer con los otros números. El ordenador seguirá estas instrucciones con total exactitud, sin cansarse y sin equivocarse (aunque reproducirá fielmente nuestros propios errores de programación), a un promedio de muchos miles de operaciones por segundo. El resultado final de estas infatigables manipulaciones no es otro que más números. Éstos son "traducidos" en la información que deseamos, de forma que nos resulten comprensibles. Es la actividad de los programadores humanos la que determina la utilidad del ordenador, explotando su destreza con los números y haciéndole desempeñar tareas que tengan algún significado para nosotros, partiendo de diversas formas de información y transformándolas según un proceso que, sin el ordenador, resultaría sumamente largo, tedioso y complicado.

Guía para adquirir un micro

Si acude a comprar un ordenador, le desconcertará la diversidad de ofertas. He aquí la guía para salir airoso de la situación

Los equipos de video, los aparatos de televisión y los sistemas de alta fidelidad poseen una característica en común: cada uno de ellos realiza una tarea específica. Entre los distintos modelos existentes sólo varía el grado de sofisticación. Pero un equipo estereofónico sólo puede reproducir sonido, una lavadora automática únicamente sirve para lavar la ropa y un televisor se limita a recibir y exhibir las señales de emisión.

Un ordenador es algo diferente. Puede que un centenar de personas compren el mismo ordenador, pero cada una de ellas le encomendará a su máquina una tarea particular propia. Por esta razón, adquirir un ordenador personal es algo completamente distinto que comprar cualquier otro aparato para el hogar.

En el momento en que usted decida comprar un ordenador personal, deberá considerar con especial cuidado diversos factores. Lo que debe hacer en primer lugar es confeccionar una lista que contenga todo lo que desearía que hiciera su ordenador. Por ejemplo, si lo que le interesa a usted es aprender los fundamentos de la programación BASIC, el ordenador más indicado sería el Sinclair Spectrum o un Oric-1. Si lo que desea es utilizar su ordenador personal como medio de juego, como procesador de textos o para manejar las cuentas domésticas, la mejor elección sería un Dragon 32 o un Commodore 64. Precio y fiabilidad son otros factores a los que deberá prestar especial atención.

Su lista de exigencias ha de ser exhaustiva, de manera que no acabe por adquirir un ordenador que en definitiva no le sirva en absoluto para lo que deseaba.

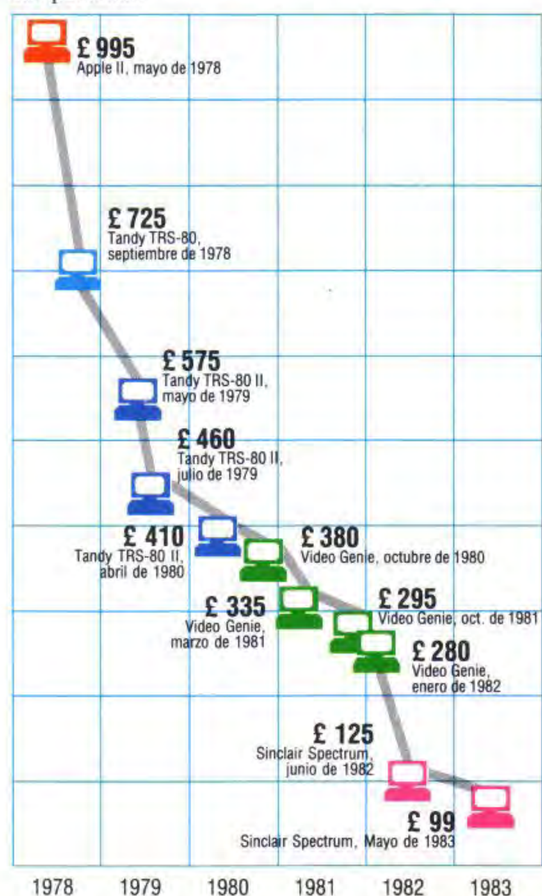
El ordenador personal elegido puede ser sólo el corazón de un sistema. Para explotar todo su potencial será necesario, asimismo, algún medio de guardar programas para su utilización en el futuro. Con este fin deberá hacer uso de una grabadora o de alguno de los sistemas de disco. Necesitará, además, un televisor para poder visualizar los programas y seguir los juegos. A menudo esto supone la compra de un segundo televisor, especialmente si usted tiene familia: ¡a ellos no les sentará nada bien perderse sus programas favoritos porque usted esté ocupado con su micro! Para todo lo que no sea relativo a juegos, también necesitará disponer de copias impresas de los programas o de los resultados producidos por el ordenador, y para ello será imprescindible contar con una impresora.

Si el ordenador se va a destinar exclusivamente a juegos, es probable que su atención se centre en la cantidad de software que haya a su disposición. En este sentido, las mayores ventajas las ofrecen micros como el Sinclair Spectrum y el Vic 20, para los que existe una variada serie de programas en cassette.

¿Cuánta memoria necesitará usted? Esto dependerá de la complejidad de los programas que desee utilizar. A menudo los programas más complejos son mayores en tamaño y, por tanto, requieren más memoria para retenerlos. Los procesadores de textos exigen grandes cantidades de memoria para su labor de almacenamiento. Por lo general, 32 K de RAM deberían satisfacer la mayoría de estas necesidades, aunque probablemente 16 K serían suficientes para llevar una programación de juegos de entretenimiento con buenos gráficos. Como regla general, decídase por una máquina que posea la mayor cantidad de memoria que usted pueda permitirse adquirir.

Algunos de los ordenadores personales más caros (como el Commodore 64 y el Atari 800) pueden considerarse aptos para su utilización en oficinas. A los dos se les pueden conectar fácilmente unidades de disco, impresoras y *modems* (dispositivos que permiten enlazar con otros ordenadores a través del teléfono).

Asegúrese, pues, de haber pensado en todos los detalles al confeccionar su lista de exigencias. Antes de desprenderse de su dinero, usted debe tener una idea clara y precisa acerca de lo que espera de su ordenador personal.



La caída de los precios del micro

Como ejemplo de la caída de los precios en los países desarrollados en el campo de la informática, mostramos un gráfico de la evolución del mercado británico para los ordenadores de precio más competitivo que tengan al menos 16 K de RAM. (Los precios no incluyen el accesorio RAM)









ATARI 400

MEMORIA NORMAL: 16 K

AMPLIABLE HASTA: No ampliable

CONECTA CON: Cassette Atari, televisor, monitor, unidad de disco Atari, impresora, palanca de mando.

VENTAJAS: Bajo costo. Amplia gama de software.

INCONVENIENTES: Teclado al tacto. El BASIC debe comprarse como accesorio.

RESUMEN: El tablero al tacto y la falta de BASIC incorporado hacen que esta máquina sea muy indicada para juegos.

ATARI 800

MEMORIA NORMAL: 48 K

AMPLIABLE HASTA: No ampliable

CONECTA CON: Cassette Atari, televisor, monitor, unidad de disco Atari, impresora, palanca de mando.

VENTAJAS: Teclado similar al de una máquina de escribir. Amplia gama de software. Programas intercambiables con el modelo ATARI 400.

INCONVENIENTES: El BASIC es demasiado prolijo y difícil. Sólo funciona con los cassettes y unidades de disco Atari.

RESUMEN: Posee un tablero mejor que el del modelo más barato, el ATARI 400, y es capaz de llevar programas tanto comerciales como de juegos.

BBC MICRO

MEMORIA NORMAL: A, 16 K; B, 32K

AMPLIABLE HASTA: 32 K

CONECTA CON: Cassette, televisor, monitor, unidad de disco, palanca de mando.

VENTAJAS: Amplia gama de software. Sonido claro al amplificarse. BASIC muy versátil.

INCONVENIENTES: Capacidad de color limitada. Sonido débil si no es amplificado. Manual de instrucciones de difícil comprensión.

RESUMEN: Esta versátil máquina puede convertirse fácilmente en un poderoso instrumento de cálculo.

COLOUR GENIE

MEMORIA NORMAL: 32 K

AMPLIABLE HASTA: No ampliable

CONECTA CON: Cassette, televisor, impresora, monitor, equipo de alta fidelidad, palanca de mando, cartuchos enchufables.

VENTAJAS: Teclado similar al de una máquina de escribir. Sonido claro.

INCONVENIENTES: Nivel de operación comparativamente lento. Gama de software limitada. Gráficos pobres.

RESUMEN: Uno de los más baratos ordenadores en color con teclado similar al de una máquina de escribir; pero para aprovechar al máximo sus cualidades debe ser utilizado por alguien experimentado.



Ian McKinnell







JUPITER ACE

MEMORIA NORMAL: 3 K+16 K

AMPLIABLE HASTA: 51 K

CONECTA CON: Cassette, televisor, monitor.

VENTAJAS: Incluye una unidad RAM de 16 K.

INCONVENIENTES: Se presenta en una caja de plástico delgado. El lenguaje FORTH no es el más indicado para un principiante. Teclado poco fiable.

RESUMEN: Apropiado para quienes deseen experimentar con FORTH.

LYNX

MEMORIA NORMAL: 48 K

AMPLIABLE HASTA: 192 K

CONECTA CON: Cassette, televisor, monitor, impresora, unidad de disco, palanca de mando.

VENTAJAS: Pantalla de visualización de datos de gran definición. Monitor de código en lenguaje máquina incorporado. Teclado similar al de una máquina de escribir.

INCONVENIENTES: BASIC muy particular. Para ampliar la memoria es el fabricante quien ha de manipular la máquina.

RESUMEN: Ordenador bien construido, con un teclado muy cómodo.

NEW BRAIN

MEMORIA NORMAL: 32 K

AMPLIABLE HASTA: 2 Mbytes (¡2 000 K!)

CONECTA CON: Cassette, televisor, monitor, unidad de disco, impresora, modem.

VENTAJAS: BASIC con muchos dispositivos. Capacidad para imprimir textos detallados. Semiportátil.

INCONVENIENTES: Clavijas bipolares no estandarizadas y muy frágiles. Demasiado complicado para un principiante.

RESUMEN: Esta máquina es sumamente ampliable y se trata de un microordenador de dimensiones reducidas altamente competitivo. Incluye una puerta (de entrada-salida) RS232.

ORIC

MEMORIA NORMAL: 16 K/48 K

AMPLIABLE HASTA: No ampliable

CONECTA CON: Cassette, televisor, monitor, modem, unidad de microdisco.

VENTAJAS: Teclado similar al de una máquina de escribir. Impresora a cuatro colores disponible. Sonido claro.

INCONVENIENTES: Ha tenido problemas de fiabilidad. Limitada gama de software.

RESUMEN: Un ordenador bien diseñado, muy adecuado para principiantes. Las teclas son estrechas y alargadas, pero resultan cómodas.





COMMODORE 64

MEMORIA NORMAL: 64 K

AMPLIABLE HASTA: No ampliable

CONECTA CON: Cassette, televisor, monitor, unidad de disco, impresora, palanca de mando.

VENTAJAS: Rápida visualización de gráficos en pantalla. Sonido claro.

INCONVENIENTES: Con frecuencia el BASIC resulta difícil.

RESUMEN: Como ordenador personal, el Commodore 64 es una buena adquisición. También puede utilizarse como máquina contable, pero para ello se requieren accesorios caros.



COMMODORE VIC 20

MEMORIA NORMAL: 5 K

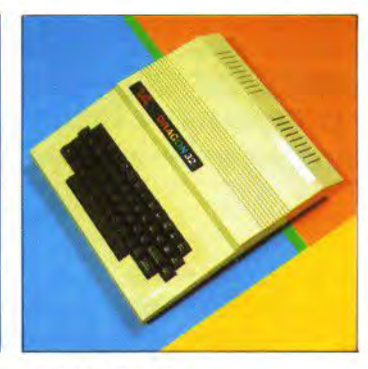
AMPLIABLE HASTA: 32 K

CONECTA CON: Cassette, televisor, monitor, unidad de disco, impresora, palanca de mando.

VENTAJAS: Teclado similar al de una máquina de escribir. Cartuchos ROM opcionales. Buena calidad de sonido. Amplia gama de software.

INCONVENIENTES: Memoria estándar pequeña.

RESUMEN: Muy popular, pero existen máquinas más baratas y de igual poder que proporcionan más memoria. Para aprovechar al máximo su potencial es necesario ampliar la memoria.



DRAGON 32

MEMORIA NORMAL: 32 K

AMPLIABLE HASTA: 64 K

CONECTA CON: Cassette, televisor, impresora, monitor, palanca de mando.

VENTAJAS: Teclado similar al de una máquina de escribir. BASIC rápido. Software por lo general intercambiable con el del ordenador Tandy Color.

INCONVENIENTES: Interruptor de corriente muy frágil. Conexión de periféricos no estándar.

RESUMEN: Sin ser una máquina excelente, es un modelo muy popular y de precio razonable.



EPSON HX20

MEMORIA NORMAL: 16 K

AMPLIABLE HASTA: 32 K

CONECTA CON: Cassette, impresora, modem, cartuchos enchufables.

VENTAJAS: Teclado similar al de una máquina de escribir. Lleva incorporado el dispositivo para visualización de datos, cassette e impresora. Amplio software y buen servicio de mantenimiento.

INCONVENIENTES: Dispositivo para visualización de datos limitado a cuatro líneas simultáneas. No parece aconsejable para un principiante.

RESUMEN: Uno de los ordenadores portátiles mejor diseñados. Incluye una puerta RS232 y una puerta de ampliación.



SINCLAIR ZX81

MEMORIA NORMAL: 1 K+16 K

AMPLIABLE HASTA: 56 K

CONECTA CON: Cassette, televisor, microimpresora.

VENTAJAS: Caracteres definibles por el usuario, lo que proporciona mayor versatilidad a la máquina. Precio muy razonable. Incluye una unidad RAM de 16 K.

INCONVENIENTES: Teclado al tacto. Su reducido tamaño puede dificultar un tanto su utilización.

RESUMEN: Este ordenador carece de sofisticación, pero puede mejorarse considerablemente recurriendo a la serie de accesorios disponibles.



SINCLAIR SPECTRUM

MEMORIA NORMAL: 16 K/48 K

AMPLIABLE HASTA: 48 K

CONECTA CON: Cassette, televisor, unidad de microdisco, modem, microimpresora, palanca de mando.

VENTAJAS: Precio muy razonable para un ordenador en color. Disponibilidad de una amplia gama de juegos y programas diversos.

INCONVENIENTES: Su reducido teclado puede ser su punto débil. Un uso prolongado puede hacer que se caliente en exceso.

RESUMEN: Muy sencillo de utilizar, cuenta con una inmensa variedad de accesorios y software. Excelente máquina para un principiante.



SORD M5

MEMORIA NORMAL: 4 K

AMPLIABLE HASTA: 36 K

CONECTA CON: Cassette, televisor, impresora, monitor, palanca de mando, cartuchos enchufables.

VENTAJAS: Bien construido. Indicador de conexión a la red eléctrica.

INCONVENIENTES: Teclas demasiado «plásticas». BASIC limitado.

RESUMEN: Ampliar esta máquina resulta demasiado caro. En el futuro se ofrecerá con un BASIC mejor y más completo.



TI 99/4A

MEMORIA NORMAL: 16 K

AMPLIABLE HASTA: 52 K

CONECTA CON: Cassette, televisor, monitor, palanca de mando.

VENTAJAS: Teclado similar al de una máquina de escribir.

INCONVENIENTES: BASIC extremadamente lento. Reducida gama de software.

RESUMEN: Se trata de una máquina poco aceptada, debido a la lentitud del BASIC y a la escasa disponibilidad de software. No obstante, su precio es bastante razonable.

Un sistema completo

Tanto para comunicarse en dos direcciones con su ordenador como para almacenar programas e incluso para algunos juegos, usted necesitará un hardware extra



Impresora

La impresora es imprescindible cuando se necesitan copias en papel de los programas o resultados impresos. Existen varias clases de impresoras, cuyo precio varía según la rapidez del proceso de impresión y la calidad de las copias

Unidades de disco

Al igual que las cassettes, las unidades o dispositivos de disco almacenan programas. En lugar de una cassette, se utiliza un "disco flexible". Las unidades de disco son mucho más caras que las grabadoras, pero trabajan mucho más rápido y almacenan mayor cantidad de información

Grabadora

Una grabadora de cassettes normal constituye una forma barata de guardar programas. El programa queda almacenado en la memoria del ordenador mientras éste lo está utilizando. Al cortarse el suministro energético, el contenido de la memoria desaparece. Antes de que esto ocurra, el programa puede grabarse en cinta magnetofónica; cuando se lo vuelva a necesitar, bastará con reproducírselo al ordenador

Televisión

Un aparato de televisión normal permite visualizar los mensajes del ordenador. Y, cuando usted escribe un programa, todo lo que vaya tecleando aparecerá reflejado en la pantalla. El monitor situado detrás del televisor está diseñado para proporcionar imágenes de mayor definición y calidad

Mando de palanca

De aspecto similar a los mandos de que disponen algunos juegos recreativos, su utilización depende del tipo de juego para el cual se emplee. Por ejemplo, puede controlar tanto una nave espacial como los caracteres de un laberinto. Algunos mandos de palanca poseen un "relleno" de diez o más botones (dispuestos como los de una calculadora), cuya utilización depende, asimismo, del juego de que se trate

Mando de bola

Este mando también se utiliza para los juegos. Haciendo rotar la bola se mueven por la pantalla los elementos del juego. Permite un ajuste de posición más preciso y más rápido que el mando de palanca y resulta más cómodo de utilizar. Está dotado de botones para tareas como disparar "láseres", etc.

El ordenador

El ordenador es el corazón de todo el sistema, aunque para poder comunicarse con el usuario requiere algunos "extras". Posee un teclado similar al de una máquina de escribir pero con algunas teclas adicionales. Cuenta con algunas clavijas bipolares (situadas, por lo general, en la parte posterior del ordenador) para conectarlo con otras máquinas, como con la grabadora o la unidad de disco, con el televisor, etc.

Mark Watkinson

Control del ordenador

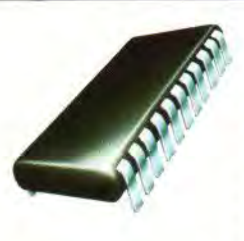
El "hardware" de su ordenador no funcionará sin la ayuda de un "software" adecuado. Conozcamos su significado y valoremos el software que se puede hallar en el mercado

El *software* constituye la mitad invisible de un sistema de ordenador. Sin él, el ordenador no es más que una masa inerte de maquinaria electrónica. Un ordenador que carezca de software no puede hacer absolutamente nada.

Si examina los chips de silicio situados en el interior de un ordenador, verá que consisten en miles, quizá millones, de interruptores electrónicos microscópicos. Así como un interruptor de luz no puede encender o apagar una lámpara por sí mismo, los interruptores de un ordenador necesitan ser encendidos o apagados. Sin embargo, no se encienden o apagan todos al mismo tiempo. Cada uno de ellos ha de ser encendido (o apagado) de manera individual y según una secuencia exacta y precisa en relación con los otros miles de interruptores. El software es la forma en que esto se lleva a cabo.

Software es el nombre con que se denomina a las instrucciones que hacen que el ordenador trabaje. Estas instrucciones asumen la forma de números que, al ser introducidos en la CPU (el corazón del ordenador; véase p. 4), preparan y restauran los interruptores internos para que éstos produzcan cosas específicas. Estos números sólo son "comprendidos" por el ordenador cuando están en el denominado sistema binario (convertidos en unos y ceros; véase p. 28).

Estos unos y ceros que el ordenador comprende (en el sentido que le hacen realizar tareas específicas) son el producto final de una larga cadena de acontecimientos que comenzaron como ideas en la mente de la persona que escribió el programa. Un programa para ordenador (la palabra "programa" designa una unidad determinada de software) puede existir de muchas maneras diferentes. Lo único que podemos afirmar en concreto de cualquier programa es que debe acabar de una forma comprensible para el ordenador. Tomemos un ejemplo específico. Supongamos que un ingeniero de tráfico desea controlar un conjunto de semáforos a través de un ordenador. Para ello, el ordenador de control necesitará de un programa que prevea la secuencia correcta de acontecimientos (¡de qué serviría que todos los semáforos se pusieran en verde al mismo tiempo!). Pero antes de que este software pueda escribirse, el ingeniero debe pensar en profundidad qué es exactamente lo que él desea que haga el ordenador. Comúnmente estas ideas se escribirían utilizando ora-





ROM

El ROM (Read Only Memory) es uno de los principales dispositivos de memoria de ordenadores. Producto de la revolución de los chips de silicio, permite almacenar con carácter permanente los programas del ordenador. La mayoría de los ordenadores personales están equipados con un chip ROM que contiene el lenguaje BASIC. En algunos el ROM puede comprarse por separado con el fin de ampliar sus prestaciones incorporándoles otro lenguaje. También existen ROMs procesadores de textos





CASSETTE

Con frecuencia el software se ofrece en cintas de cassette, idénticas a las que se emplean para grabar sonido. Los programas de juegos suelen ofrecerse en esta forma. El programa se pasa de la cinta al ordenador conectando la máquina a una grabadora de cassette normal y reproduciendo (botón "play") la cassette con el programa. Cuando el programa ha sido cargado, la cinta se detiene y generalmente el ordenador ya no necesita volver a "leerla"







DISCO FLEXIBLE

El software (los programas) puede almacenarse grabándolo en un disco de película magnética. La grabación se realiza sobre "pistas" en la superficie, similares a los surcos de un LP común, mediante una cabeza magnética de "lectura y escritura" que también "lee" (reproduce) el programa al ser requerida. Los discos ofrecen enorme capacidad y alta velocidad de operación, pero requieren sofisticadas "unidades de disco" (ver p. 8)





CARTUCHO

Un cartucho es, básicamente, un ROM convenientemente empaquetado. Algunos ordenadores poseen conectores de fácil acceso en los cuales pueden enchufarse los cartuchos. El software que se ofrece en cartuchos suele incluir un lenguaje de programación (como el BASIC) o bien sofisticados programas de juegos

Para que su ordenador funcione, es necesario "alimentarlo" con software (una serie de instrucciones electrónicas). Los dispositivos que aquí se describen son los "medios" que sirven para almacenar dichas instrucciones. Representan las cuatro formas más comunes en las que se proporciona el software. Cada una de ellas ofrece ciertas ventajas particulares. El software se confecciona a la medida de cada ordenador; el programa escrito para un ordenador determinado no siempre puede ser utilizado con otro

ciones corrientes. (Por ejemplo: "En este momento quiero que el semáforo número uno cambie a ámbar manteniendo encendida la luz roja. Luego quiero que tanto la luz roja como la luz ámbar se apaguen y que se encienda la verde".) Es evidente que estas oraciones no están escritas de forma comprensible para ningún ordenador y, por tanto, han de ser convertidas en un programa. El ingeniero utiliza un lenguaje de programación, como el BASIC. Un lenguaje de este tipo permite reescribir los pensamientos ordenados lógicamente (en español) de forma que el intérprete BASIC los pueda comprender. El intérprete BASIC es, en sí mismo, un programa que convierte el programa original (escrito en BASIC) en la forma que le resulte comprensible a la unidad central de proceso (CPU). El software de esta forma se denomina "lenguaje máquina" o "código en lenguaje máquina".

El software que usted compra para utilizar con su ordenador siempre está en lenguaje máquina y es almacenado en una forma rápidamente accesible al ordenador. Algunas veces el software se almacena en el interior del ordenador, en la memoria ROM. Lo más común es que se ofrezca en cassette o en disco flexible. Estos objetos no constituyen en sí mismos el software, sino que son el "medio" a través del cual se proporciona el software. Para ser utilizado por el ordenador, el software ha de ser transferido desde la cassette (o desde el disco flexible o el ROM) al ordenador. Una vez cargadas estas instrucciones (así se denomina el proceso de transferencia de software), el programa puede comenzar a funcionar.

Cómo comprar software

Si usted tuviera en el banco medio millón de pesetas, quizá se diría: "Creo que voy a comprarme un coche". Es poco probable que alguien en esa situación se dijera: "Creo que voy a comprarme una máquina", porque la pregunta obvia sería: "¿Qué clase de máquina? ¿Que sirva para qué?"

Con el software ocurre lo mismo. Un ordenador es, en sí mismo, un objeto inerte, pero el software que usted compre para utilizar con el ordenador es capaz de convertirlo en una máquina de juegos, en una máquina de escribir automática o en un experto contable doméstico. De modo que lo primero que ha de decidir es qué función quiere que realice el ordenador.

Comience por determinar el problema y luego busque el software que le proporcione las soluciones. En la búsqueda del programa adecuado, a medida que usted analice sus necesidades reales se irá produciendo un proceso de decantación natural. Si el punto de partida es cómo entretener a los niños un domingo por la tarde, la etapa siguiente será la de buscar algún tipo de programa que con toda seguridad les proporcione ese entretenimiento. Los juegos para ordenador comprenden desde "masacres" de seres extraterrestres hasta complicadas y emocionantes simulaciones fantásticas (véase p. 32). Si lo que usted desea proporcionarle al ordenador es un programa de juegos en serie, la pregunta siguiente será si existen o no para su máquina.

Debido a que las diferencias existentes entre un ordenador y otro no se refieren sólo a su aspecto (cada ordenador posee en su interior una electrónica propia y requiere un software escrito especialmente para él), todos los modelos son prácticamente incompatibles. Un programa que funcione con el Atari 800 no servirá para el Spectrum (a menos que se fabrique una nueva



FIRMWARE

El origen del término "hardware" es obvio: se trata de la parte física y electrónica del ordenador (las conexiones para la alimentación eléctrica, teclado, chips de silicio, etc.). La palabra "software" sugiere, en cambio, una naturaleza intangible, y designa simplemente a una serie de instrucciones. Los especialistas en ordenadores hablan también de "firmware". En las primeras épocas de las computadoras, el software se codificaba y almacenaba en cintas de papel perforadas similares a las que emplean los operadores de télex. Luego las cintas de papel fueron reemplazadas por cassettes y discos magnéticos. En los años 70 se inventó una nueva técnica para almacenar software en chips ROM, diseñados a tal fin (véase p. 4). A los chips ROM se les incorporan las instrucciones de software en el proceso de fabricación. Se denomina "firmware" a la combinación de software "intangible" y hardware "tangible"



Editar



Tratamiento de textos

Con un programa de tratamiento de textos, su ordenador le ofrece avances notables en relación a la máquina de escribir convencional. Hasta los más eficientes mecanógrafos suelen cometer errores, pero con un procesador de textos usted puede conseguir cartas perfectas impresas y, al mismo tiempo, aumentar la productividad.

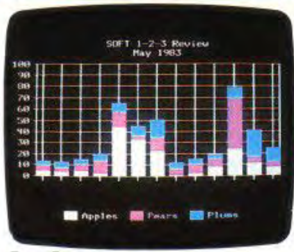
El teclado del ordenador sustituye a las teclas de la máquina de escribir. Las palabras que usted teclea aparecen instantáneamente en la pantalla, así como aparecerían, con la máquina de escribir, sobre el papel. Pero aquí termina toda similitud y entra en juego el potencial del ordenador.

En la pantalla los errores pueden corregirse de forma instantánea. Las palabras se reescriben o se hacen desaparecer. Incluso pueden borrarse párrafos enteros. Sin embargo, los procesadores de textos no sólo sirven para borrar palabras. Si sus pensamientos quedaran expresados con mayor claridad reordenando las oraciones, usted puede hacerlo ahí mismo, en la pantalla. Las palabras u oraciones que desea desplazar a otro sitio de la "hoja" se borran temporalmente (el programa de tratamiento de textos las retira de la pantalla y las almacena en la memoria del ordenador). Luego usted puede insertarlas exactamente donde lo desee.

Una vez redactado el documento en la forma que usted desea, puede imprimirse utilizando la impresora del ordenador, o almacenarse en una cassette o en un disco flexible para uso ulterior

¿QUE DESEO QUE REALICE MI ORDENADOR?

Llevar libros de cuentas



Labores contables

Si consideramos que los ordenadores pueden realizar operaciones matemáticas, no debemos sorprendernos de que existan muchos programas para ayudar al hombre de negocios. La gama de programas de gestión contable es impresionante, desde teneduría de libros automatizada hasta balances completos. Por lo general los programas de este tipo deben manejar grandes cantidades de información y necesitan almacenar un elevado número de documentos. Por este motivo suelen requerir al menos una unidad de disco flexible para hacer frente a tal demanda de almacenamiento.

Los programas contables trabajan generalmente a través de un sistema de preguntas (visualizadas en la pantalla del ordenador) y respuestas (proporcionadas por el operador del ordenador). La información que teclea el operador es manipulada por el programa del ordenador; hechos todos los cálculos, los resultados se almacenan en el disco flexible o se imprimen mediante la impresora.

Este tipo de programas incluyen la expedición automática de facturas, reestructuración de stock y confección de libros mayores, mientras se controla el trabajo en curso. Este software suele ser caro, pero puede constituir una buena inversión para el empresario, puesto que reduce el costo del trabajo y ofrece rápidos resultados

Archivo



Base de datos

Los ordenadores pueden buscar información en los archivos mucho más rápido que una persona; cuanto mayor sea la cantidad de información entre la cual usted ha de buscar un dato, mayor es la ayuda que puede proporcionarle un ordenador. En su forma más simple (y más barata), una base de datos es poco más que una agenda computerizada que puede buscar nombres, direcciones y números de teléfono. Los programas de base de datos más sofisticados y más caros pueden realizar operaciones mucho más complicadas.

Para formarnos una idea del potencial de una base de datos, tomemos el ejemplo de un botánico que esté recogiendo información para un libro que tratará sobre setas exóticas y venenosas. Este botánico habrá confeccionado extensos ficheros acerca de las diversas especies y su medio natural. También habrá tomado apuntes extraídos de diversos libros de consulta y tendrá una lista bibliográfica interminable.

Antes de que el ordenador se convirtiera en una máquina asequible, la información recogida debía ser transcrita en fichas, que tenían que ser ordenadas alfabéticamente en un fichero. Con un programa de base de datos y un ordenador, toda la información puede almacenarse en la memoria de éste. Valiéndose del potencial de la base de datos, el botánico puede obtener al instante la respuesta a sus consultas. Si necesitara confeccionar una lista de todas las setas que existen en Cataluña, la base de datos podría hacerla por él. Si precisara una lista alfabética de todos los libros que contengan las palabras "veneno" o "venenoso" y "hongo" o "seta", también la base de datos trabajaría por él.

Por lo general, las bases de datos sólo se ofrecen en discos flexibles

Manejar números



Hojas electrónicas

La "hoja electrónica" es la respuesta a todas esas preguntas como "¿y qué pasaría si...?", que suelen resolverse con una calculadora y varias bobinas de papel. Toda empresa que tiene un producto para vender posee muchas variables. La modificación de una de ellas afecta por lo general a las otras.

Consideremos las preguntas que puede formularse el propietario de un cine: "¿Si vendiera todas las localidades, a cuánto podría rebajar el precio de cada localidad?"; "¿Obtendríamos mayores ingresos bajando el precio de los helados y manteniendo la misma cantidad de vendedores, o sería más rentable aumentar el precio y tomar dos empleados nuevos?". Es probable que ambas decisiones afecten a todo el negocio; el descenso de los precios podría implicar mayores ventas, pero reducir el margen de beneficios. Una hoja electrónica es un programa especial que puede ofrecer respuestas inmediatas para este tipo de planteamientos.

Todos los números básicos que han de manejarse se disponen en forma de líneas y columnas y se especifica la relación existente entre cada línea y cada columna (por ejemplo: los números de cada línea de la columna C son el resultado de restar al número de la columna A el número de la columna B). Una vez se ha reunido toda la información real y la hipotética, cada una de las cifras puede ser alterada y la "incidencia" de este cambio en las otras cifras queda instantáneamente a la vista.

Los usuarios de hojas electrónicas suelen ser ejecutivos responsables de costos o ingenieros y científicos que deben trabajar con información numérica variable. Por lo general, las hojas electrónicas requieren unidades de disco y una impresora

Diversión



Juegos

Los ordenadores no sólo son excelentes para el tratamiento de números y de textos. También pueden proporcionar muchas horas de diversión si se utilizan con uno de los muchos programas de juegos disponibles. Éstos abarcan una amplia gama que va desde juegos de mesa, como el ajedrez y el backgammon, hasta los juegos de los salones recreativos (como el "aterrizaje lunar" y los que simulan vuelos). También existen juegos de aventuras muy complejos que pueden durar días y hasta semanas (véase p. 14). Muchos juegos para ordenador tienen también un interesante valor educativo.

Los juegos para ordenador requieren una atención y participación constante por parte del jugador. Por lo general esta participación se realiza a través del teclado; una tecla puede utilizarse para disparar un "láser" o un "misil", o para controlar el movimiento de algo que aparezca en la pantalla. El número de teclas a emplear dependerá del juego elegido y del nivel de control que exija el programa.

Una alternativa al teclado muy extendida es la palanca de mando. Estos dispositivos se enchufan en el ordenador y, en cierto modo, operan como las palancas de mando de los aviones. Proporcionan un mayor control y hacen los juegos por ordenador mucho más atractivos

Programas listos para utilizar

La mayoría de los programas que existen en el mercado se venden en disco o en cassette con un manual de instrucciones para su uso. El Apple Writer es un clásico programa de tratamiento de textos. Consta de un único disco flexible y de un manual en el que se explica cómo utilizarlo. El manual incluye un cursillo para que los principiantes puedan comenzar a usarlo de inmediato. Los niveles de documentación varían enormemente. Algunos manuales son tan incompletos y están tan mal escritos, que la utilización del programa resulta difícil y hasta imposible. Este es un factor a tener en cuenta al comprar un programa

versión especial del Spectrum); de manera que usted debe comprar aquellos programas que hayan sido producidos especialmente para su ordenador.

Pero aún no está usted preparado para hacer su compra. Las consideraciones siguientes se refieren a las limitaciones físicas de su máquina. Verifique cuánta memoria posee su ordenador. Si tiene 16 K de ROM, compruebe si el juego que usted desea requiere que se le agregue memoria. Por regla general, los juegos más interesantes y más sofisticados exigen programas más largos, por lo cual necesitará que su ordenador disponga de mayor memoria. Y no olvide que el software se ofrece en distintos soportes (véase página anterior). Si un programa sólo está disponible en disco flexible y usted posee únicamente cassette, no podrá utilizarlo si antes no adquiere una unidad de disco. Algunos programas (en especial los de juegos) requieren también otros accesorios, por ejemplo, palanca de mando. Probablemente lo que no necesite para sus juegos sea la impresora, pero los programas de oficina a menudo sí la necesitan para imprimir los resultados.

Por último, está lo concerniente a cuánto dinero puede usted gastar. Los precios de ciertos juegos presentados en cassette varían notablemente. Algunos paquetes de gestión en disco suelen ser caros.



Fausto Dorelli

Tipos de software

En cierto modo, los juegos constituyen una clase aparte. Al fin y al cabo, la función de un juego es la de entretener. La mayoría de los otros programas están diseñados para hacer que un trabajo determinado se realice más fácil y rápidamente. En este aspecto, los informáticos han logrado aumentar la rentabilidad y la eficacia de muchísimas maneras. Pensemos en la pobre mecanógrafa cuyas copias no satisfacen al jefe. Con un microordenador y un programa de tratamiento de textos, el ordenador reemplaza a la mecanógrafa y todas las correcciones se hacen en pantalla. Una vez que todas las palabras se han escrito en la forma correcta, la página completa puede imprimirse en papel automáticamente y con sólo pulsar una tecla. Se obtendrán así valiosos resultados en cuanto a economía de tiempo y se evitarán frustraciones.

Otro trabajo agotador que se presta a la computación es la administración financiera. Muchas de las actividades que solían mantener ocupados a ejércitos de empleados (calculando salarios y actualizando los libros de la empresa) pueden ahora llevarse a cabo mediante software escrito con esa finalidad. Los programas en sí mismos son bastante especializados, por lo cual es poco probable que un único programa satisfaga todas las necesidades de una oficina. Las categorías incluyen programas de "nóminas" para calcular los salarios e imprimir las hojas de recibo, programas de "control de stock" para mantener actualizada la relación de los productos vendidos o utilizados (en algunos casos el programa puede encargarse de solicitar nuevos stocks automáticamente) e incluso existen programas para ayudar a decidir el tamaño y la calidad de papel más económicos para imprimir libros o revistas.

Otras tareas en las que los ordenadores alcanzan cotas espectaculares de eficacia son las de archivo y selección de información. Este tipo de programas se denominan "base de datos". Las bases de datos pueden reemplazar muebles archivadores completos y realizar toda la labor de ordenación y clasificación.

La última categoría general de software es la que se conoce como "hoja electrónica". Un programa de hoja electrónica permite confeccionar y modificar cuantas veces se desee complicados presupuestos o previsiones financieras, en lugar de tener que recurrir a la calculadora y a las bobinas de papel.

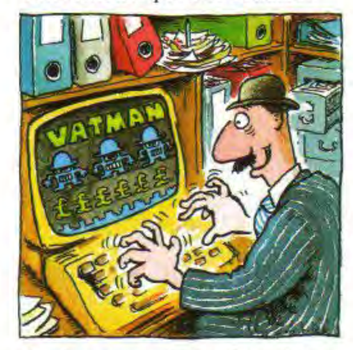
Todos los tipos de software a los que nos hemos referido se venden "hechos". Están "listos para utilizar" en el sentido que el informático tenía en mente una serie específica de soluciones para los problemas que alguien le había planteado. No obstante, puede llegar el día en que ningún programa de los disponibles en el comercio haga que su ordenador realice exactamente la función para la cual usted lo necesita. ¿Y qué hará usted en ese caso? Una solución, si bien bastante onerosa, consiste en "alquilar" a un programador de ordenadores para que le escriba un programa hecho exactamente a la medida de sus necesidades. La otra solución consiste en que usted mismo aprenda a escribir sus propios programas. Si llega a dominar un lenguaje como el BASIC, le será posible producir programas que hagan que su ordenador realice toda clase de cosas sorprendentes.



Preguntas y respuestas

Preguntas muy corrientes sobre ordenadores a las que no suelen dar respuesta los manuales y las revistas especializadas

¿Cuál es la utilidad de un ordenador personal, además del entretenimiento que proporcionan los juegos?
Los ordenadores personales se utilizan para la administración de pequeños comercios, manejo de cuentas, tratamiento de textos e incluso para mantener actualizado el registro de los socios de un club o ayudar en el diseño de interiores. Éstos son algunos de los usos de los ordenadores personales. De un modo más generalizado, estos usos pueden clasificarse en manejo de números, manejo de palabras, almacenamiento de información y visualización de la misma, en la forma más conveniente para el usuario.



Según parece, los ordenadores son cada vez más pequeños y más baratos. ¿Dónde y cuándo se detendrá este proceso?
En la medida en que la tecnología continúe avanzando, los ordenadores irán disminuyendo de tamaño. Los componentes electrónicos interiores cada vez son más pequeños, pero las dimensiones del teclado no pueden alterarse de forma radical. Éste seguirá siendo lo suficientemente grande mientras debamos utilizarlo con nuestros dedos. Por esta causa, es poco probable que aparezcan ordenadores mucho más pequeños, a menos que se reemplace el teclado tradicional por otros sistemas.

En cuanto al precio, y teniendo en cuenta que hay ordenadores en el mercado por menos de 20 000 pesetas, es poco probable que se abaraten mucho más.



¿Resulta dificultosa la comprensión del lenguaje BASIC?
El aprendizaje del BASIC en sí mismo no resulta en absoluto difícil. Si lo comparamos con el castellano, que, en definitiva, todos hemos debido aprender, el BASIC es un lenguaje sencillo con reglas fijas. Por esta causa su aprendizaje resulta mucho más sencillo que el de una lengua extranjera. Sin embargo, pese a que el aprendizaje del BASIC es fácil, lo que ya no resulta tan sencillo es escribir programas largos y complicados.

¿Cuándo debería empezar el aprendizaje del BASIC?
El mejor momento para empezar es cuando usted lo necesite. En algunos casos, el ordenador podrá realizar exactamente lo que usted desea sólo cargándolo con un programa comprado. En estos casos, no es necesario que aprenda BASIC. Lamentablemente, los programas que usted pueda comprar no siempre se ajustarán con toda exactitud a lo que desea. En algunas ocasiones, basta con que aprenda un poco de BASIC para que pueda adaptarlos a sus necesidades. No obstante, para que el ordenador haga lo que usted desee y sea realmente un ordenador personal, debe comenzar el aprendizaje del BASIC lo antes posible.

Los monitores suelen ser algo más caros que los televisores. ¿Qué beneficio supone esta diferencia adicional?
En la pantalla de un monitor, las imágenes aparecen mucho más claras y mejor definidas. Si usted proyecta pasarse cierto tiempo utilizando su ordenador, gran parte de ese tiempo lo invertirá en observar la pantalla; y si en la misma aparecen imágenes nítidas y de fácil lectura, probablemente finalizará esta actividad sin que le duela la cabeza. Además, cualquier imagen gráfica que usted cree con el ordenador podrá verla mejor en la pantalla de un monitor.

¿Cuánto consume un ordenador personal?
Menos que una bombilla de 60 vatios.



De acuerdo con lo que se afirma en los anuncios publicitarios, ¿es cierto que puedo utilizar el ordenador para contribuir a la educación de mis hijos?
Sí. Y también puede ayudar en la educación de los adultos. El Departamento de Educación y Ciencia de Gran Bretaña, por ejemplo, ha hecho un gran esfuerzo para que todas las escuelas del país dispongan de un ordenador, lo que induce a afirmar que los ordenadores están llamados a desempeñar un papel de creciente importancia en la educación. La clave de su utilización como un medio para la enseñanza radica en la disponibilidad de un software apropiado. Por ejemplo, existen muchos programas de ejercicios para enseñar ortografía y la tabla de multiplicar. Sin embargo, ésta no es una manera muy estimulante para la utilización del ordenador. Una aplicación pedagógica mucho más imaginativa es la que se realiza a través del lenguaje LOGO. El LOGO permite que los niños aprendan mediante la exploración del llamado micromundo y la realización de experimentos para comprobar lo que ocurre. En este caso, el aprendizaje del niño se lleva a cabo programando realmente el ordenador. (Véase p. 34.)



Algunos ordenadores, como el Spectrum, poseen teclados de plástico similares al de una calculadora; otros, como el Dragon, tienen teclados similares al de una máquina de escribir. ¿Cuál es la diferencia entre ambos?
Mínima, a menos que usted sea un experto mecanógrafo; en este caso, podrá escribir sus programas BASIC a mayor velocidad en un teclado similar al de una máquina de escribir que en uno tipo calculadora.

Los juegos favoritos

Mazmorras y dragones, bolsa de valores y vuelos espaciales, emoción y educación: todo está en los juegos por ordenador

1



Muchas personas utilizan sus ordenadores para jugar. El mundo de los juegos por ordenador es un fascinante caleidoscopio de emoción, problemas de ingenio y novedosos desafíos.

Tanto en las salas recreativas como en el hogar, las máquinas de videojuegos están siendo superadas por ordenadores personales, más económicos y más potentes, que le ofrecen mayor variedad y emoción sin obligarlo a incurrir en gastos excesivos mientras aprende a manejarlos. Estos juegos para ordenador personal han hecho que muchas personas descubrieran las posibilidades de diversión del ordenador, algo que no han logrado nunca (ni lograrán en el futuro) los programas de gestión empresarial.

Aventuras

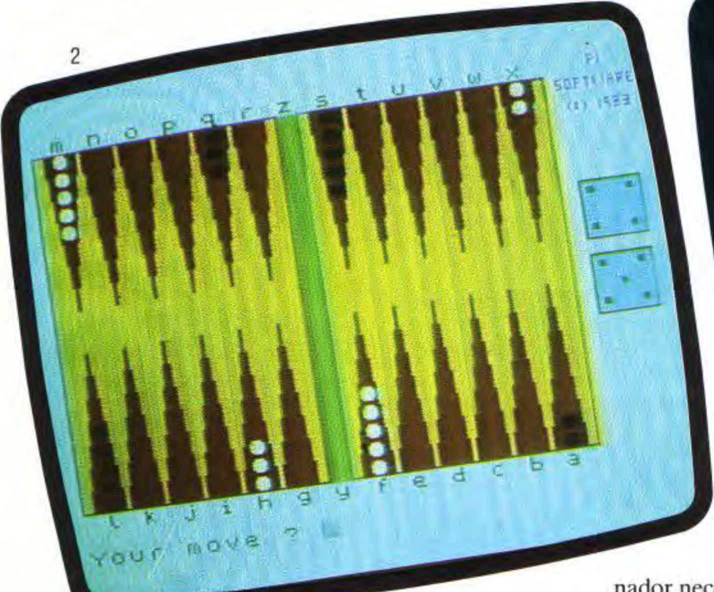
Para que un juego sea apasionante no es necesario que tenga imágenes en color y sonido. La divulgación del potencial de los ordenadores baratos ha dado lugar a la aparición de una nueva gama de juegos; se trata de juegos que estimulan la imaginación con palabras, tal como lo han hecho los libros desde siempre.

Estos juegos se llaman "de aventuras", debido a que el primer programa fue escrito para que los programadores lo jugaran en sus ordenadores gigantes en su tiempo libre. La idea consiste en que el programa crea un mundo que debe ser explorado por el jugador guiando un *alter ego* (otro yo); pero esta labor de exploración se realiza por medio de palabras que se escriben en el teclado, no a través de palancas de mando. En el mundo interior del ordenador, el *alter ego* se desplaza escribiendo en el teclado, por ejemplo, "norte" y "arriba", y el ordenador proporciona con palabras las imágenes que describen los alrededores y todos los objetos circundantes. El pequeño personaje puede "coger", "arrojar", "hacer girar" o "romper" los objetos, o emprender cualquier acción que considere conveniente. El jugador puede intentarlo todo. El único límite es el de su imaginación.

El mundo interior del programa del ordenador puede ser un laberinto de cuevas y mazmorras habitado por monstruos y con tesoros escondidos, como en el juego *Dungeons and dragons* (Calabozos y dragones). O puede ser una nave espacial abandonada por extraterrestres o incluso una casa de campo en la que se ha cometido un crimen que debe ser esclarecido.

Cualquiera que sea el escenario, el jugador ha de explorarlo, descubrir objetos útiles o tesoros y resolver problemas de tipo intelectual. El mundo del ordenador necesita estar reflejado en un mapa, y los resultados completos sólo se proporcionan una vez que todos los escollos hayan sido superados y descubiertos todos los tesoros. Los mejores juegos de aventuras pueden equipararse a una buena novela, y llegar a su final puede ocupar más tiempo que leer un libro.

2



3

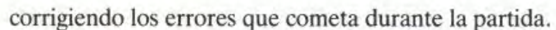
4

Juegos de mesa

Era natural que, tan pronto como el nivel de tecnología lo permitiera, los juegos de mesa favoritos en todas las épocas se transfirieran a los ordenadores personales. No es necesario buscar un adversario, pues el ordenador desempeña a la perfección ese papel. Y si usted realiza un mal movimiento, puede corregirlo sin que por ello el ordenador lo acuse de tramposo. La máquina puede incluso mejorar su juego señalando y

6





corrigiendo los errores que cometa durante la partida.

El ajedrez por ordenador ha alcanzado un nivel muy alto y los gráficos del ordenador personal ahora pueden producir trazados de tablero completos y detallados, con suave movimiento de las piezas. El juego de damas ha sido analizado exhaustivamente, y con toda seguridad un programa para ordenador podría erigirse en campeón del mundo. *Backgammon, bridge, Othello, Pontoon* (el veintiuno) y *Go,* todos están en el mercado para diversas máquinas personales.

Aprender jugando

Algunas empresas de software se especializan en juegos educativos en los cuales el ordenador sugiere ejercicios y problemas, y ofrece, a cambio de una respuesta correcta, la recompensa de un juego. Puede suceder también que el juego tenga el aspecto de tal pero posea, asimismo, una información educativa entremezclada en él, como sumar o deletrear.

Uno de estos juegos consiste en que el niño guía, sobre una hoja de papel, a un robot que sostiene un lápiz y que por su forma recuerda a una tortuga. De este modo los niños realizan divertidas imágenes dibujadas y, al mismo tiempo, aprenden geometría.

Juegos recreativos

Son juegos de acción y movimiento rápido y han recogido miles de millones de monedas a través de las ranuras de las máquinas de video. Con un ordenador personal usted tiene a su alcance todos los juegos recreativos que desee: invasores atacantes, ranas saltarinas, mineros que excavan y gorilas gigantes.

Pero las empresas de software tienen sus propias ideas y han creado juegos aún superiores a aquéllos en emoción y espectacularidad de los gráficos. Al disponer de un ordenador personal, se amplía la gama de juegos rápidos y electrizantes; y usted puede divertirse cuando lo desee y sin necesidad de introducir por la ranura cantidades ingentes de monedas.

Alta estrategia

También los juegos que requieren pensar y planificar con la complejidad y resolución de un gran general han entrado en los dominios del micro.

El jugador puede convertirse en cualquier general de cualquier guerra, desplegando ejércitos e intentando intuir y anular los planes del enemigo. El ordenador está en su elemento actuando como árbitro y manipulador del tablero, ya que hace innecesarias las detalladas reglas escritas y las numerosas piezas de cartón, causantes hasta ahora de que los juegos de guerra estuvieran reservados sólo a una minoría.

Otra posibilidad es que el jugador sea el rey de un pequeño país y que se dedique a administrar las cosechas y el tesoro mientras mantiene felices y bien alimentados a sus súbditos, cuidando de que nadie robe en sus dominios.

5



1 Hechicería: Penetre en un mundo misterioso. Sus compañeros pueden ayudarlo. Pero ¿a quién elegir? ¿Un guerrero, una doncella o un científico?

2 Backgammon: En este juego su adversario está respaldado por una potencia formidable: la inexorable lógica del ordenador

3 ABC Dragon: Un juego educativo para niños

4 Zaxxon: Uno de los juegos recreativos más originales. ¡La pantalla se convierte en el parabrisas del piloto mientras éste se abre camino hacia su objetivo, eludiendo misiles y refriegas aéreas!

5 Frente del Este: Tal vez usted consiga triunfar en Rusia, donde fracasó Hitler

6 Simulacro de vuelo: ¡Haga volar el avión o se estrellará!

El jugador puede tomar bajo su responsabilidad las reservas energéticas de un país, ponderando los costos de las mismas en relación a sus peligros y efectos a largo plazo. El ordenador puede proporcionarle mayor amplitud de miras y perfeccionar y profundizar sus puntos de vista sobre la situación mundial.

Vuelo de altura

Los programas de juegos pueden introducirlo a usted en la cabina de un avión ligero, permitiéndole leer los instrumentos y manejar los controles para realizar despegues y aterrizajes perfectos; pueden convertirlo en piloto de una misión de la lanzadera espacial, haciéndolo disfrutar incluso de imágenes que muestran a la Tierra vista desde las ventanillas; o en un magnate de las finanzas que opere en el mercado de valores de todo el mundo, influyendo en el crecimiento o causando la bancarrota de gigantescas empresas.

La exactitud lo es todo en los juegos de simulacro. Siga las reglas del mundo verdadero y el juego le mostrará lo que ocurriría realmente. Pero no se preocupe si comete usted algún error, porque no se encontrará en medio de un naufragio ni tendrá que arrojarse desde la ventana de un edificio de Wall Street. ¡Los ordenadores saben perdonar!

La letra menuda

No se puede juzgar un libro por su cubierta. Pero podemos saber mucho acerca de un micro por sus especificaciones técnicas

Teclado

El teclado está diseñado especialmente para que su utilización resulte cómoda y sencilla. Sus teclas presentan una disposición estándar, similar a la de una máquina de escribir. El carácter de cada tecla puede visualizarse repetidamente manteniendo pulsada la misma. Cada letra puede visualizarse en mayúscula o en minúscula y para los números existe un grupo separado de teclas (relleno de teclas numéricas)

Visualización de datos

El juego de caracteres ASCII (American Standard Code for Information Interchange, código estándar utilizado en los Estados Unidos para intercambio de información) es un juego convencional de números, letras y símbolos que utilizan muchos ordenadores. En algunos modelos, la pantalla exhibe estos caracteres en 80 columnas y 25 líneas. La imagen puede mostrarse en un televisor o en un monitor especial

Memoria

Los números indican la capacidad de memoria en kilobytes o miles de bytes. La ROM (Read Only Memory, memoria de lectura solamente) posee los medios que se requieren para el funcionamiento básico del ordenador, incluyendo por lo general un lenguaje como el BASIC. La RAM (Random Access Memory, memoria de acceso directo) sirve para el almacenamiento de los programas y de la información del usuario

Interfaces

Existen enchufes especiales mediante los cuales puede conectarse al ordenador una impresora, un equipo de comunicaciones, una grabadora y cartuchos. Un cartucho es un ROM especial que puede contener un programa, un lenguaje e incluso un nuevo O/S (sistema operativo)

CPU

La CPU es la unidad central de proceso (Central Processing Unit), chip de silicio que constituye el corazón del ordenador. Éste, un microprocesador Zilog Z80, es uno de los más comunes. El reloj que regula sus operaciones es capaz de medir con una exactitud de 2,2 millones de veces por segundo

Características de un ordenador típico

Memoria	16 Kbytes de ROM, 32 Kbytes de RAM, capaz de asignar dirección a 48 Kbytes de RAM
Visualización de datos	Puede visualizar juego de caracteres ASCII de 25 líneas con 80 posiciones de caracteres, salidas a aparato de televisión y monitor
CPU	Z80 funcionando a 2,2 MHz
Teclado	Diseño ergonómico, teclado QWERTY, medios de repetición, relleno de teclas numéricas superior e inferior
Interfaces	Interface impresora, interface para comunicaciones, conexión cassette, interface de cartucho
BASIC	Órdenes para gráficos y para sonido, verificación de sintaxis, mensajes de error, vuelco de memoria en pantalla
Gráficos	Compatibles con teletexto y videotexto, máxima resolución 640×256, efecto tridimensional
Sonido	Sintetizador de música, 5 octavas, salida para alta fidelidad
Periféricos disponibles	Unidad de cassette, unidades de disco flexible, unidad de disco rígido, impresoras, trazador de gráficos, digitalizador, palanca de mando, modem, sintetizador de voz
Lenguajes disponibles	FORTH, PASCAL, LOGO, LISP, PROLOG, ASSEMBLER

BASIC

El lenguaje incorporado de un ordenador suministra las órdenes que permiten la utilización de los recursos de gráficos y de sonido. Controla las instrucciones que le han sido proporcionadas al ordenador para corroborar que éstas sean correctas: si se ha cometido un error, emite un mensaje de error. El vuelco de memoria en pantalla reproduce la pantalla en la impresora. Se proporcionan órdenes extras de BASIC para asegurar que los programas estén escritos con una buena "estructura", es decir, que sean fáciles de leer y corregir

Sonido

Pueden ejecutarse notas o acordes individuales dentro de una escala de cinco octavas, y la señal de sonido puede emitirse a través de un sistema de alta fidelidad

Gráficos

Las visualizaciones producidas por teletexto y videotexto pueden ser exhibidas en la pantalla, que contiene 256 líneas, cada una con 640 puntos, para visualización de gráficos. Se pueden producir y exhibir imágenes en perspectiva de objetos tridimensionales

Periféricos disponibles

Las unidades que pueden acoplarse al ordenador incluyen: grabadora, unidad de disco flexible y unidad de disco rígido. Los tres almacenan programas e información. Para producir palabras e imágenes puede usarse una impresora matricial o una impresora de tipos, así como un trazador de gráficos y un digitalizador para la entrada y salida; para juegos puede acoplarse una palanca de mando. Un modem es un dispositivo que permite a los ordenadores comunicarse por teléfono

Lenguajes disponibles

En lugar del BASIC pueden utilizarse estos lenguajes de ordenador; cada uno de ellos está pensado para un tipo determinado de aplicación. El ASSEMBLER es un tipo de lenguaje de programación más difícil que, por ejemplo, el BASIC, pero consigue que los programas se lleven mucho más rápido

Sinclair Spectrum

Los soberbios gráficos en color y su exclusivo sistema de almacenamiento en microdisco, a un precio sorprendentemente bajo, son las principales características del Spectrum. Pero... atención al teclado

El Sinclair Spectrum es un pequeño ordenador personal con gráficos en color, capacidad para la producción de sonido y una gran memoria. Su éxito se explica, sobre todo, por su bajo precio, que ha dado como resultado un gran volumen de ventas. Esta demanda, a su vez, ha desembocado en la creación de empresas que se dedican a producir programas para ser llevados con el Spectrum y a la fabricación de accesorios susceptibles de acoplársele.

El Spectrum ofrece al usuario una gran capacidad de memoria, en realidad hasta 48 K, de modo que puede actuar como vehículo para programas largos escritos en el propio lenguaje de ordenador BASIC de la máquina. Estos programas pueden estar ya escritos (almacenados en cassettes o en discos) o bien escribirlos los mismos usuarios del Spectrum. La versión BASIC del Spectrum, al igual que la de casi todos los otros ordenadores personales, posee sus propias características, pero se acerca lo suficiente a la versión estándar del BASIC, aceptada mayoritariamente, como para resultarle familiar a cualquier programador de este lenguaje.

Los recursos del Spectrum para la emisión de sonidos y para la confección de gráficos pueden controlarse mediante el BASIC. El control de sonido del Spectrum es bastante primitivo, realizándose a través de la orden denominada BEEP. Con algo de habilidad, puede lograrse que produzca un pequeño repertorio de efectos sonoros. Por el contrario, su capacidad para la creación de gráficos en color es realmente impresionante. Permite una visualización de gráficos cuyas configuraciones van desde las órdenes PAPER (papel) e INK (tinta) para el control de los colores de fondo y de primer plano, hasta órdenes para el trazado de círculos y para hacer que zonas determinadas de la pantalla se enciendan y se apaguen de forma intermitente.

Los programas ya escritos que existen en el mercado para el Spectrum incluyen una inmensa variedad de juegos. Éstos abarcan desde juegos ya clásicos, como los "invasores del espacio" y *Pac-Man*, hasta juegos de aventuras y de simulaciones de vuelo, que asombran por su originalidad.

La capacidad de ampliación del Spectrum ha sido satisfecha, hasta cierto punto, por la propia Sinclair, gracias a la inclusión de una impresora y un sistema de almacenamiento: la impresora ZX y el microdisco ZX. Sin embargo, muchos otros fabricantes ofrecen accesorios para esta máquina, como palancas de mando, interfaces (con las cuales el ordenador puede controlar otros equipos o bien comunicarse con ellos), teclados similares al de una máquina de escribir y sintetizadores de sonido y de voz.

El teclado Sinclair



El Spectrum posee un económico teclado con piezas de movimiento limitado. Sinclair optó por mejorar el teclado del ZX81, pero sin incurrir en los costes excepcionales que hubiese implicado un teclado totalmente móvil.

El teclado del Spectrum utiliza una pieza de plástico de línea sencilla para incorporar las teclas, que sobresalen de la cubierta exterior. Al pulsar una tecla, ésta cierra un contacto situado debajo de ella. De este modo, el ordenador reconoce que una pieza ha sido pulsada y pone en funcionamiento el carácter correspondiente. El medio para hacer que la tecla pulsada vuelva a levantarse no es otro que la elasticidad de la hoja de plástico, que se estira al teclear la pieza.

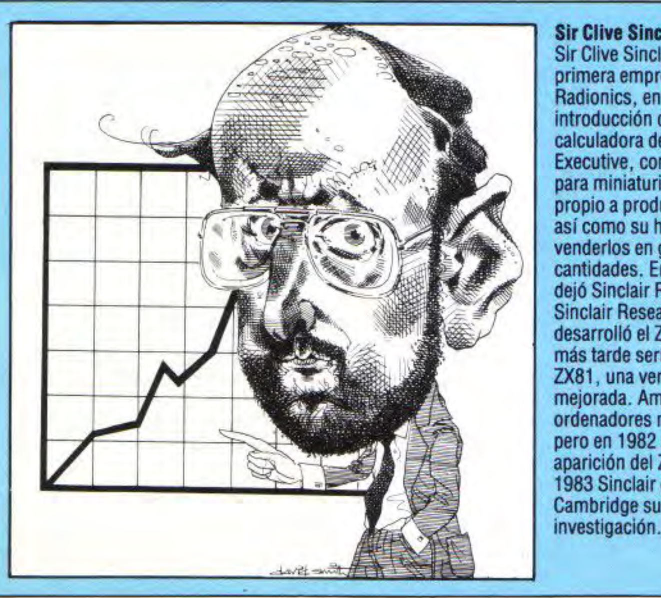
Esta técnica de diseño ha reducido el costo del teclado y ha contribuido al bajo precio del Spectrum



El microdisco ZX
El microdisco ZX conecta con el Spectrum a través de una interface acoplada en la parte posterior de la base de la máquina. Proporciona 100 Kbytes de almacenamiento; el tiempo promedio requerido para el acceso a los datos es de 3,5 segundos



La impresora ZX
La impresora ZX se enchufa directamente en el Spectrum. Imprime nueve líneas de texto por pulgada. La impresora también produce gráficos mediante la impresión de los caracteres para gráficos. El contenido de la pantalla puede ser copiado mediante la impresora utilizando la orden COPY



Sir Clive Sinclair
Sir Clive Sinclair fundó su primera empresa, Sinclair Radionics, en 1962. En 1972, la introducción de la primera calculadora de bolsillo, la Executive, confirmó su aptitud para miniaturizar y dar un estilo propio a productos populares, así como su habilidad para venderlos en grandes cantidades. En 1979 sir Clive dejó Sinclair Radionics y fundó Sinclair Research. En 1980 desarrolló el ZX80, que un año más tarde sería seguido por el ZX81, una versión modificada y mejorada. Ambos eran ordenadores monocromáticos, pero en 1982 se produjo la aparición del ZX Spectrum. En 1983 Sinclair estableció en Cambridge su propio centro de investigación.

Reloj
Este reloj electrónico marca el compás 3,5 millones de veces por segundo para dar sincronía a las operaciones que realiza el Spectrum

Chip video
El circuito integrado produce una señal en color que es enviada al modulador para que se visualice en una pantalla de televisión

16 K de RAM
Ésta es la memoria del Spectrum. Los 16 K se consiguen con chips de 2 K de memoria dispuestos en una hilera de 8

Memoria disponible
Ésta es la memoria que se proporciona con el ordenador para que el programador almacene programas e información. De las capacidades estándar, la menor es de 16 Kbytes y la mayor de 48 Kbytes

Disipador
Esta gran placa de aluminio disipa en forma de calor toda la energía no deseada. Después de estar en actividad durante un tiempo prolongado, el Spectrum se calienta, lo cual indica que esta placa está cumpliendo su función

Conector del tablero
Aquí se enchufa el teclado en el ordenador principal

Modulador
Recoge la señal en color del chip vídeo y la convierte en una señal del mismo tipo para producir su visualización, del mismo modo en que lo haría una antena de televisión

Conectores de micrófono
Estos conectores se utilizan con una grabadora, para transferir al ordenador la información proveniente del cassette y viceversa

Chip input/output
Adapta de modo adecuado las entradas del teclado y la unidad de cassette para que el ordenador pueda utilizarlas, y modifica convenientemente la información proveniente del ordenador para que sea visualizada en la pantalla

Conector de alimentación
Aquí se conecta al ordenador la alimentación eléctrica de nueve voltios proveniente de la unidad de alimentación eléctrica del Spectrum

Conector marginal
Aquí es donde se acoplan al Spectrum los periféricos, como la impresora ZX

Microprocesador
Éste es el motor del ordenador en el corazón del Spectrum. Es un microprocesador Zilog Z80A

Chip básico
Este chip proporciona el lenguaje de programación BASIC del Spectrum. Es una ROM de 16 Kbytes. El chip examina las instrucciones de programación dadas al Spectrum y las traduce del modo adecuado para que el microprocesador pueda ejecutarlas

Altavoz
El pequeño zumbador eléctrico genera los sonidos que puede producir el Spectrum

Regulador de voltaje
Recibe el suministro de nueve voltios provenientes del enchufe red y los convierte en los niveles requeridos por los diversos componentes electrónicos del Spectrum

SINCLAIR SPECTRUM

DIMENSIONES
232×144×30 mm

PESO
552 g

CPU
Z80A

VELOCIDAD DEL RELOJ
3,5 MHz

MEMORIA
RAM de 16 Kbytes ampliable hasta 48 Kbytes. ROM de 16 Kbytes, con BASIC incorporado

VISUALIZACION EN VIDEO
24 líneas con 32 posiciones de caracteres, o 192×256 puntos para gráficos de gran resolución. Ocho colores en ambos casos

INTERFACES
Conector de televisión, conector de cassette (sin control remoto), conector de 28 patillas para la conexión de periféricos

LENGUAJE PROPORCIONADO
BASIC

OTROS LENGUAJES DISPONIBLES
FORTH, PASCAL, LISP, LOGO, PROLOG

VIENE CON
Unidad de alimentación eléctrica (sin enchufe), conector de antena TV, conector de cassette, cassette de demostración, 2 manuales

TECLADO
40 teclas móviles, todas en una única hoja de plástico

DOCUMENTACION
Un breve manual de presentación describe la manera de instalar el ordenador. Otro, más voluminoso, trata de la programación BASIC. Este último comienza con una descripción del teclado que debería ser más detallada, lo que ayudaría al principiante a conocer mejor el modo de utilizar las teclas. Los capítulos dedicados a programación BASIC demuestran las posibilidades del Spectrum a través de numerosos ejemplos de instrucciones individuales y algunos programas cortos. Una serie de apéndices ofrecen una guía de referencia bastante completa acerca del Spectrum y de su BASIC. Los manuales están bien elaborados y proporcionan una buena información acerca de cómo instalar el Spectrum y del potencial de la máquina

Obedeciendo órdenes

Su ordenador hará exactamente lo que usted desee cuando le "hable" en la forma correcta... y él no se equivocará nunca

Otros lenguajes

El lenguaje más utilizado en la mayoría de microordenadores es el BASIC. Pero de ninguna manera el BASIC es el único lenguaje. Antes de que aparecieran los microordenadores, cuando la mayoría de los cálculos se realizaban en ordenadores de unidad principal que ocupan cuartos enteros, los científicos y los ingenieros empleaban un lenguaje denominado FORTRAN. En el mundo de los micros, otros lenguajes que gozan de popularidad son el PASCAL, el FORTH y el LOGO

Pascal

Al igual que el BASIC, el PASCAL fue desarrollado fundamentalmente como un lenguaje de enseñanza para los estudiantes de programación. Es muy apreciado por los profesores de informática, puesto que favorece la escritura de programas sofisticados y muy bien urdidos. Por lo general el PASCAL se ofrece en discos flexibles y suele ser caro. Para el Spectrum existe una versión económica en cassette

Forth

Los programas escritos en lenguaje FORTH se asemejan mucho menos al idioma inglés que el BASIC o el PASCAL. El FORTH también resulta más difícil de aprender. Sin embargo, tiene la ventaja de poseer una mayor riqueza de expresión, puesto que en unas pocas líneas pueden escribirse complicados programas. Con el FORTH usted puede definir sus propias órdenes, mientras que, en el caso del BASIC, éstas ya vienen predefinidas

Logo

El LOGO es un lenguaje relativamente nuevo que está popularizándose en el campo de la educación. Tiene la gran ventaja de que es lo suficientemente simple como para que incluso los niños pequeños puedan aprenderlo. Puede ayudar a enseñar técnicas de programación y también favorece un acercamiento lógico al diseño de programas desde un primer nivel. El LOGO utiliza gráficos "tortuga" que permiten que las imágenes aparezcan rápidamente en pantalla. También se puede conectar al ordenador una tortuga mecánica (véase p. 34). Al pulsar sencillas órdenes en el teclado, la tortuga se mueve y dibuja líneas y formas

Es perfectamente factible que cualquier persona utilice un ordenador (en su casa o en el trabajo) sin saber nada en absoluto acerca de cómo funciona. Partiendo de este punto, iniciamos un curso que explica paso a paso, desde el principio, todo lo que usted necesita saber para crear con éxito sus propios programas.

Al cabo de cierto tiempo muchas personas descubren que los programas y los juegos que han comprado para su ordenador empiezan a resultarles un poco aburridos, y se preguntan si pueden modificarlos o incluso escribirse sus propios programas. Pero un ordenador no puede hacer nada por sí mismo. Se le debe proporcionar una lista de instrucciones que le informen con todo detalle qué es exactamente lo que debe hacer y qué pasos dar para poder conseguirlo. Estas instrucciones conforman lo que se denomina un *programa,* y el arte de crear un programa se conoce como *programación*.

La programación no es particularmente difícil. No requiere siquiera que usted tenga aptitud para las matemáticas, a menos, por supuesto, que desee escribir programas para llevar a cabo tareas matemáticas. En principio, lo único que necesita es comprender BASIC.

Su primer lenguaje

Muchos ordenadores personales traen incorporado un lenguaje denominado BASIC. Como su nombre indica, está diseñado para que los principiantes aprendan las nociones básicas de la programación de forma rápida y sencilla. Como cualquier lenguaje humano, el BASIC posee una gramática, un vocabulario y una sintaxis propios, si bien el vocabulario es muchísimo más reducido que el de un idioma normal. El BASIC utiliza varios vocablos ingleses cortos, fáciles de reconocer y sencillos de aprender. En términos generales, el BASIC es adecuado tanto para el principiante como para el usuario más experimentado.

No obstante, uno de los inconvenientes de este lenguaje radica en que, a través de los años, los distintos fabricantes de ordenadores han ido incluyendo sus propias modificaciones. Como resultado de ello existe una gran cantidad de variantes del BASIC, particularmente respecto a las órdenes para controlar los aspectos de la máquina de desarrollo más reciente, por ejemplo, color, gráficos y sonidos. Todas las variaciones del BASIC que se producen en los ordenadores más populares se muestran en cada lección, en el recuadro "Complementos al BASIC".

Debido a las variaciones que experimenta el BASIC de un ordenador a otro, es prácticamente imposible escribir un programa en BASIC que pueda llevarse con cualquier ordenador. No obstante, afortunadamente el lenguaje posee un núcleo común, que por lo general es el mismo en todas las máquinas. Comenzaremos centrándonos en ese núcleo y, a medida que el curso avance, iremos introduciendo programas más complicados.

Los primeros pasos

Empecemos por escribir un pequeño programa y veamos lo que sucede. Éste nos mostrará cómo el ordenador comete, aparentemente, un error. Encienda el ordenador y digite el programa tal y como se lo presentamos, incluyendo todos los espacios. El <CR> al final de cada línea es para recordarle que digite *Carriage Return* (retorno de carro). En su ordenador, esta tecla puede estar señalizada como RETURN, ENTER o incluso con el signo ↵.

```
10 REM LOS ORDENADORES NUNCA SE
   EQUIVOCAN <CR>
20 PRINT "DIGITE UN NUMERO" <CR>
30 INPUT A <CR>
40 LET A = A + 1 <CR>
50 PRINT "CREO QUE EL NUMERO
   DIGITADO ERA"; <CR>
60 PRINT A <CR>
70 END <CR>
```

Después de haber tecleado todo lo anterior, pulse LIST <CR>. El programa que usted acaba de digitar debería reaparecer en la pantalla. LIST es una instrucción para que el ordenador "imprima" un listado del programa en memoria. Si el programa aparece correctamente en pantalla después de digitar LIST, podemos tratar de hacerlo funcionar (RUN). No se preocupe si comete algún error al pulsar el programa. Después de haber listado (LIST) el programa, simplemente vuelva a digitar la línea que contenga el error. No se olvide del número de la línea. Pruebe tecleando

```
25 REM AQUI HAY OTRA LINEA 'REM'<CR>
```

y luego digite (LIST) el programa otra vez. Para librarse de la línea, pulse sólo el número de línea, seguido de <CR>. Cuando esté satisfecho y el programa haya sido digitado correctamente, puede "ponerlo en funcionamiento" digitando RUN<CR>. Inténtelo y verá aparecer en pantalla:

```
DIGITE UN NUMERO
```

Siga adelante y pulse un número. Pruebe con el 7 (utilice los guarismos; recuerde que el ordenador no reconocería que "siete" es 7 a menos que lo programemos especialmente para ello). Después de apretar 7, en la pantalla se leerá:

```
CREO QUE EL NUMERO QUE HA DIGITADO ERA 8
```

¿Se ha equivocado realmente el ordenador, o simplemente estaba obedeciendo órdenes? Si examinamos el programa línea por línea podremos ver cada una de las instrucciones que le hemos proporcionado al ordenador. He aquí la primera línea:

```
10 REM LOS ORDENADORES NUNCA SE EQUIVOCAN
```

REM es la abreviatura de REMark (observación). Todo lo que aparece en la misma línea después de REM es ignorado por el ordenador. Las observaciones constituyen una manera muy práctica de hacernos recordar lo que está haciendo el ordenador. Esta REM en particular no es más que un título, no nos indica lo que está haciendo el programa. Ya avanzado el curso, veremos la gran ayuda que representan unas REM bien escritas. Ahora echemos una mirada a:

```
20 PRINT "DIGITE UN NUMERO"
```

Cuando el BASIC llegue a la palabra PRINT, todo lo que sigue a continuación es "impreso" en la pantalla del ordenador. Observe que la oración está entre comillas dobles. Una de las reglas del BASIC es que los caracteres (letras) que aparecen entre comillas dobles después de la enunciación PRINT aparecerán en pantalla exactamente como fueron digitados. En la línea 60 veremos otra forma de utilizar PRINT. A continuación sigue:

```
30 INPUT A
```

Por ahora pasaremos esta línea por alto; volveremos a ella después de ocuparnos de la línea 40.

```
40 LET A = A + 1
```

Aquí la letra A está utilizada como una variable. Una variable es como una caja etiquetada que puede contener tanto un número como algunos caracteres. En lugar de tener que recordar qué contiene la caja, lo que tenemos que saber es cómo se llama la caja para poder referirnos a ella. Es algo así como decir "Alcánceme la caja que lleva la etiqueta B" en lugar de decir "Alcánceme la caja que contiene los tornillos de 15 mm de cabeza".

En esta línea tenemos una «caja» denominada A. Esta caja se llama variable porque el valor de lo que pongamos en ella puede variar. A una variable podemos asignarle virtualmente cualquier valor. En la línea 30 le habíamos asignado un valor a la variable A, así que volvamos a ella:

```
30 INPUT A
```

La utilización de la palabra INPUT es, en BASIC, uno de los modos de asignarle (otorgarle) a una variable un valor específico. Cuando el programa BASIC llega a una línea que comienza con INPUT, espera a que algo sea digitado en el teclado. INPUT A permite que el ordenador sepa que tenemos una variable denominada A y que lo que se digite en el teclado será el valor asignado a esa variable. En este punto, digitando 7<CR> coloca 7 en la caja A o, utilizando términos informáticos, asigna el valor 7 a la variable A. Ahora que ya sabemos en qué consiste una variable y que conocemos uno de los modos de asignarle un valor, volvamos a ocuparnos de la línea 40.

```
40 LET A = A + 1
```

El nombre de la variable a la cual se le asigna un valor aparece siempre a la izquierda del signo igual. Aquí le estamos dando a A un nuevo valor. La sentencia significa "Hagamos (LET) que el nuevo valor de A sea igual al antiguo valor más 1". El antiguo valor de A era 7. Ahora lo hemos convertido en 7 + 1, de modo que el nuevo valor es 8.

```
50 PRINT "CREO QUE EL NUMERO DIGITADO ERA";
```

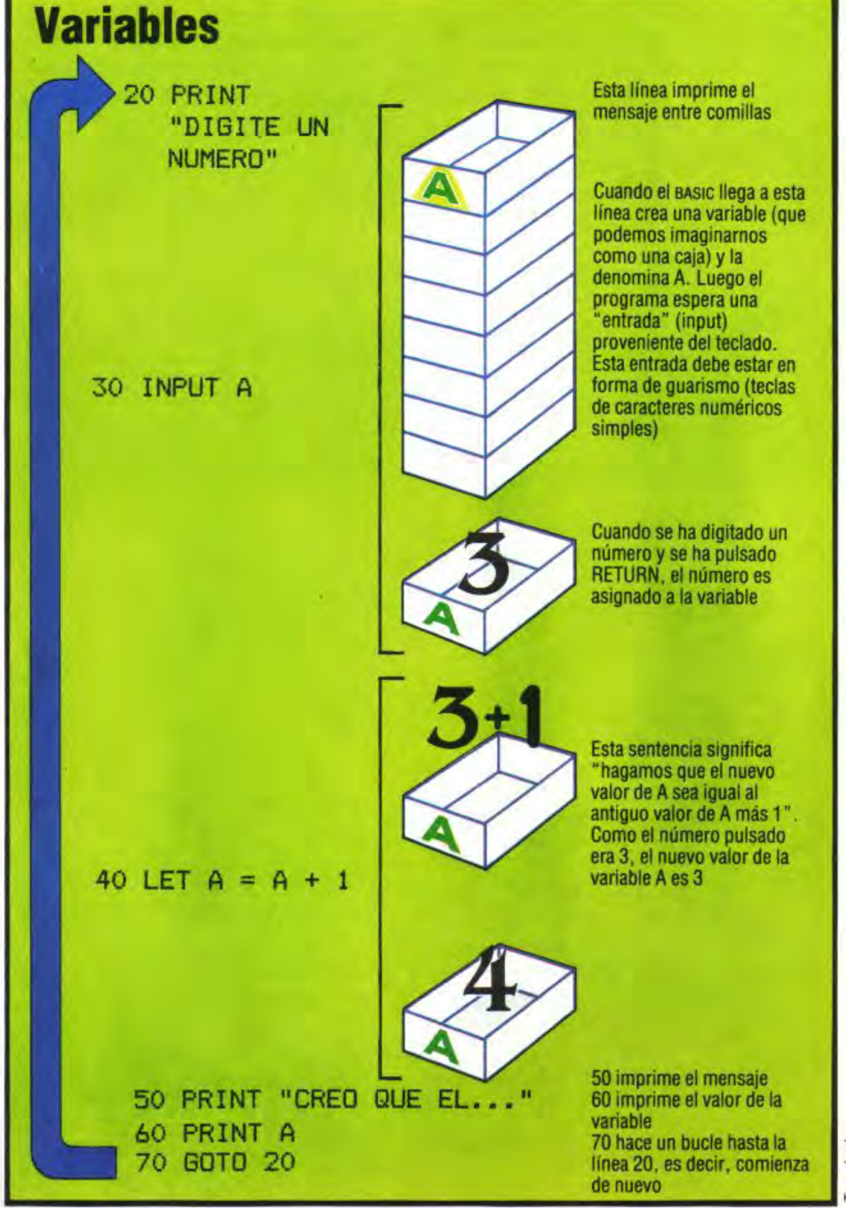
Ésta es, nuevamente, nuestra sentencia de impresión. Esta sentencia "imprime" la secuencia de caracteres (es decir, las palabras o los números que usted ha digitado) entre las comillas dobles. Observemos el punto y coma al final de la línea. Éste ayuda a especificar las posiciones en las cuales las cosas se imprimen en pantalla. Ya avanzados en el curso, retomaremos con mayor detalle el uso del punto y coma. Ahora ocupémonos de:

```
60 PRINT A
```

Aquí hay otra sentencia PRINT, pero esta vez A no está entre comillas. Ya sabemos que el programa no imprimirá en pantalla una A verdadera, porque hemos visto ya que para ello se requiere el entrecomillado. Sin las comillas, el BASIC busca una variable que posea la misma etiqueta que el carácter que sigue a PRINT. Si lo encuentra, imprime el valor de la variable. (¡De no encontrarlo, proporcionaría un mensaje de error!) Este programa ya posee una variable denominada A y por ello el BASIC imprime su valor. ¿Y cuál es ese valor?

Si usted cree que la respuesta es 7, recuerde que el BASIC trabaja con los programas línea a línea, siguiendo el orden de los números de línea. Llegados a la

El recuadro inferior muestra cómo se utilizan las "variables" en BASIC. También ilustra cómo se utiliza la sentencia GOTO (véase página siguiente) para formar un bucle



Complementos al BASIC



Sólo el Sinclair Spectrum utiliza el elemento LET de la instrucción. En otros ordenadores, éste está implícito, es decir, puede pasarse por alto. Por ejemplo, la línea 20 se podría escribir A = A + 1 en lugar de LET A = A + 1



Este elemento no se utiliza en el Spectrum. Se supone que la última línea digitada del programa es el final de éste



En el Spectrum aparece en la pantalla como dos palabras separadas (GO TO), aunque sólo se oprime una tecla. La mayoría de los otros ordenadores aceptan que la instrucción se digite en dos palabras

línea 60, el valor de A ya había sido cambiado a 8, y eso es lo que el ordenador imprimirá. Finalmente llegamos a:

```
70 END
```

La sentencia END (fin) le dice al BASIC que se ha llegado al final del programa. Algunas versiones de BASIC insisten en que todos los programas deben terminar en END, mientras que otras no (ver el recuadro "Complementos al BASIC").

Observe que cuando usted pone en funcionamiento el programa, éste sólo lo hace una vez. Para que funcione otra vez, debe volver a digitar RUN<CR>.

Veamos ahora una manera de hacer funcionar el programa todas las veces que deseemos utilizando la sentencia GOTO.

Utilizando GOTO

A continuación proporcionamos el mismo programa pero con una línea adicional. Si usted había apagado el ordenador para tomar un descanso, digítelo. De no ser así, bastará con teclear las líneas 70 y 80. Éstas aparecen en azul en el siguiente listado.

```
10 REM LOS ORDENADORES NUNCA SE
   EQUIVOCAN<CR>
20 PRINT "DIGITE UN NUMERO"<CR>
30 INPUT A<CR>
40 LET A = A + 1<CR>
50 PRINT "CREO QUE EL NUMERO
   DIGITADO ERA";<CR>
60 PRINT A<CR>
70 GOTO 20<CR>
80 END<CR>
```

Después de haberlo digitado y listado (LIST), trate de imaginarse lo que sucederá antes de intentar llevarlo (RUN). Luego pulse RUN<CR> y, al igual que en la primera versión del programa, aparecerá:

```
DIGITE UN NUMERO
```

Digite cualquier número (utilizando las teclas numéricas) y oprima RETURN. El ordenador le sumará 1 al número y lo expondrá al final del mensaje.

```
CREO QUE EL NUMERO DIGITADO ERA 8
```

Esto es seguido inmediatamente por un nuevo mensaje DIGITE UN NUMERO. Digitando otro número y volviendo a apretar RETURN, el programa repetirá este ciclo *ad infinitum*. La razón para que se produzca esta repetición la hallaremos en la línea 70:

```
70 GOTO 20
```

Cuando el BASIC llega a una sentencia GOTO, en lugar de continuar con la línea siguiente, va hacia (GOes TO) la línea cuyo número se especifica. En este caso se dirige nuevamente hasta la línea 20 y hace que todo el programa sea repetido otra vez. Y así continuaría indefinidamente realizando estos bucles. Si usted desea dejar de llevar el programa, se encontrará con que no existe manera de interrumpir estos bucles. El programa sigue adelante esperando su entrada.

Como usted comprenderá, existen maneras de escribir el programa que permitan pararlo cuando lo deseemos, y de ellas nos ocuparemos en el próximo fascículo. Mientras, deberemos detener el programa. Si el ordenador utilizado tiene una tecla BREAK, puede usarla para hacer que el programa deje de funcionar. Digitando RUN<CR> el programa comenzará nuevamente.

Observe que aún tenemos la expresión END al final del programa. Tal como hemos escrito este programa, con la sentencia GOTO 20 creando un bucle interminable, nunca llegaremos hasta el final, ¡pero algunas versiones de BASIC insisten en que utilicemos siempre un END al final!

Si no encuentra una manera de detener el programa, inténtelo oprimiendo la tecla RESET. Con ello es prácticamente seguro que el programa se detendrá. Luego pulse nuevamente LIST. Si obtiene un listado, usted podrá "editar" el programa en los ejercicios que proporcionamos a continuación. Si no obtiene un listado, ello significa que el RESET de su ordenador destruye el programa en memoria y, por lo tanto, usted deberá digitarlo completo otra vez.

Ejercicios

Estos ejercicios han sido cuidadosamente seleccionados por grados y están concebidos para que resulten amenos. Intentar resolverlos es una de las mejores maneras de verificar si usted ha comprendido el material que le hemos presentado y si efectivamente avanza en sus conocimientos de informática.

Antes de comenzar con los ejercicios, trate de modificar unas pocas líneas para ver qué efectos producen estos cambios en la forma de llevar el programa. No existe ninguna posibilidad de que el ordenador resulte dañado, aunque usted cometa errores o pulse teclas equivocadas. Para cambiar una línea, digite el programa y luego verifique el resultado haciendo el listado (LIST). En la pantalla volverá a aparecer todo el programa. Teclee el número de la línea que desea cambiar, seguido de la nueva línea. Intente con esta sentencia:

```
10 REM LOS ORDENADORES ALGUNAS VECES SE
   EQUIVOCAN<CR>
```

luego digite LIST otra vez. Observe cómo ha cambiado la primera línea. Si desea borrar toda la línea, basta con que pulse el número de línea seguido de <CR>. Pruebe:

```
10<CR>
LIST
```

La línea 10 debería desaparecer. Ponga de nuevo la línea 10 volviendo a digitar toda la línea al completo; ¡y no se olvide del número de línea!

■ Reescriba el programa de modo que el ordenador imprima realmente el número digitado. Una pista: el truco está en quitar una línea entera.

■ Vuelva a digitar la línea 70, de modo que el programa pase a la línea 80. Haga el listado del programa (LIST). Haga funcionar el programa (RUN). ¿Por qué el programa no se ha comportado de la misma manera que antes?

■ Modifique la línea 60 para que el ordenador imprima en pantalla una A en lugar del valor de la variable A.

■ Reescriba la línea 60 de modo que el ordenador vuelva a imprimir el valor de la variable A. Quite por completo la línea 10 (la línea de REM). Digite RUN. ¿Hay alguna diferencia en la forma en que se lleva el programa?

■ Incluya una nueva observación (REM) en la línea 25. Pueden agregarse nuevas líneas con sólo digitar el nuevo número seguido de la nueva sentencia. Coloque una observación en la línea 25 para recordarle lo que ocurrirá luego; podría ser algo así como "espera una entrada proveniente del teclado". Después de digitar la nueva línea y de pulsar <CR>, vuelva a listar (LIST) el programa y verifique que su nueva observación aparezca en el lugar correcto.

■ Reescriba el programa de modo que multiplique por 10 el número que usted digite. Necesitará modificar la línea 50 para imprimir algo parecido a EL NUMERO QUE USTED DIGITO MULTIPLICADO POR 10 ES. En esta ocasión no deseamos sumar al valor de la antigua variable, sino multiplicarlo por 10. Para designar "multiplicado por", el BASIC utiliza el signo *. (No utilice una X, porque el BASIC sólo reconoce en ella a una letra y no al signo de multiplicación).

Hemos recorrido ya un gran trecho. Hemos visto cómo escribir comentarios, que para el BASIC son observaciones (REMarks), cómo imprimir (PRINT) secuencias de caracteres en la pantalla, cómo imprimir (PRINT) el valor de una variable en pantalla y cómo hacer que el programa vaya hacia (GOTO) un número de línea determinado.

El próximo tema del que nos ocuparemos es cómo salir de un bucle utilizando una expresión IF-THEN. Descubriremos cómo hacer que el programa "realice" para nosotros un número de ciclos determinados en lugar de seguir haciendo bucles de manera indefinida.

Y entonces surgió el BASIC

En la actualidad, el BASIC es el lenguaje de programación más popular del mundo. Los lenguajes de ordenador se inventaron para que el operador humano pudiera comunicarse más fácilmente con la máquina, y el BASIC es uno de los lenguajes más sencillos de aprender y de utilizar. Consiste en una serie de instrucciones simples en inglés combinadas, cuando es necesario, con los símbolos matemáticos del teclado de una máquina de escribir.

El BASIC es un lenguaje al que se llega a dominar rápidamente. A los pocos minutos de haber desembalado un microordenador, usted puede estar escribiendo programas sencillos. Fue desarrollado en 1965 en el Dartmouth College de New Hampshire (Estados Unidos), con el único fin de simplificar los lenguajes ya existentes. Sus inventores fueron dos maestros, Thomas Kurtz y John Kemeny. El uso universal del BASIC ha supuesto la introducción en el mismo de ligeras variaciones de lenguaje. Pero el núcleo del lenguaje BASIC sigue siendo respetado por todos los fabricantes.

Un programa es una secuencia de instrucciones que el ordenador ejecuta para llevar a cabo una tarea específica. La tarea puede ser realizar una previsión financiera mensual o trasladar a un "invasor del espacio" a través de la pantalla del televisor. El programa aparece como una serie de líneas numeradas. Las órdenes se aprenden rápidamente y hasta el programa más complicado no utiliza más que combinaciones y repeticiones de las órdenes elementales.

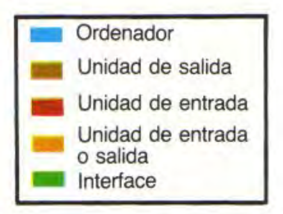


El BASIC ha desmitificado la programación y ha hecho del ordenador algo accesible a cualquier persona

La mayoría de los ordenadores salen de fábrica con el BASIC incorporado. Los ordenadores también pueden programarse en "código de lenguaje máquina" (al que se describe como un lenguaje de "bajo nivel", porque su estructura se asemeja a la lógica de los circuitos electrónicos). El BASIC es un lenguaje de "alto nivel" porque estructuralmente se aproxima al inglés cotidiano. Para aplicaciones técnicas y especializadas existen muchos otros lenguajes de alto nivel, pero el BASIC es el lenguaje idóneo para poder introducir todos los demás. Es un lenguaje sencillo y poderoso.

La revolución mundial

La revolución de los ordenadores se extiende por toda la Tierra y está cambiando los hábitos de la sociedad. Es el mundo del mañana, pero se vive hoy



La revolución de los ordenadores coincidió con la carrera espacial y con el viaje que llevó al primer hombre a la Luna. Los millones de dólares que se invirtieron en este esfuerzo tuvieron como consecuencia directa la concentración de las mejores mentes científicas y la apertura de una brecha en cuanto a técnicas de fabricación. El viaje a la Luna hizo posible algo casi imposible. Los efectos inmediatos abarcan desde la creación de nuevos materiales de cerámica, plásticos y adhesivos hasta la microminiaturización de un increíble potencial informático.

Cuando un "ordenador" se identificaba con bastidores llenos de circuitos, también implicaba miles de componentes electrónicos unidos entre sí mediante cables individuales. Los costos de fabricación eran enormes. Ahora esos mismos circuitos pueden introducirse en chips de silicio lo suficientemente pequeños, como para caber en una carcasa del tamaño de un teclado.

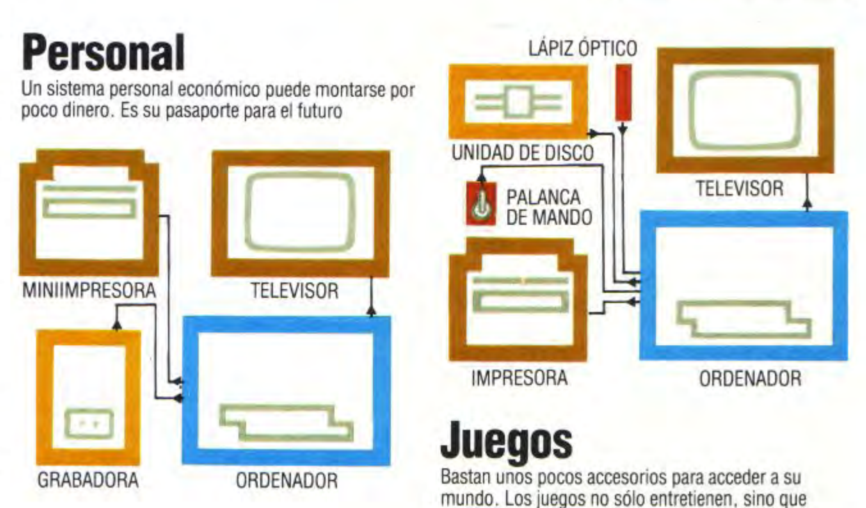
Los chips albergados en el interior de ordenadores como el Dragon o el Spectrum no sólo son pequeños, sino que pueden fabricarse en serie a un costo casi insignificante. El silicio, la materia prima a partir de la cual se producen los chips, es uno de los elementos más comunes de la Tierra. Los granos de arena están compuestos en su mayor parte por silicio.

Ahora que el ordenador ha conquistado un sitio en los hogares y en las oficinas de todo el mundo desarrollado, tenemos la oportunidad de constituirnos en privilegiados testigos del comienzo de la segunda revolución industrial. La primera revolución industrial sustituyó la mano de obra del trabajador por una maquinaria motorizada. La revolución del ordenador supondrá un ahorro del tiempo de trabajo del personal especializado y reemplazará a los obreros de las fábricas por robots controlados por ordenador.

La incidencia a largo plazo de esta revolución en nuestras vidas no está ni mucho menos esclarecida. Lo único que puede suponerse con toda certeza es que los patrones de trabajo y ocio sufrirán modificaciones y que éstas se producirán con rapidez. Los robots, extensiones mecánicas de los ordenadores, están reemplazando a los obreros. Éstos se enfrentan a la disyuntiva de especializarse en algo o quedarse sin trabajo.

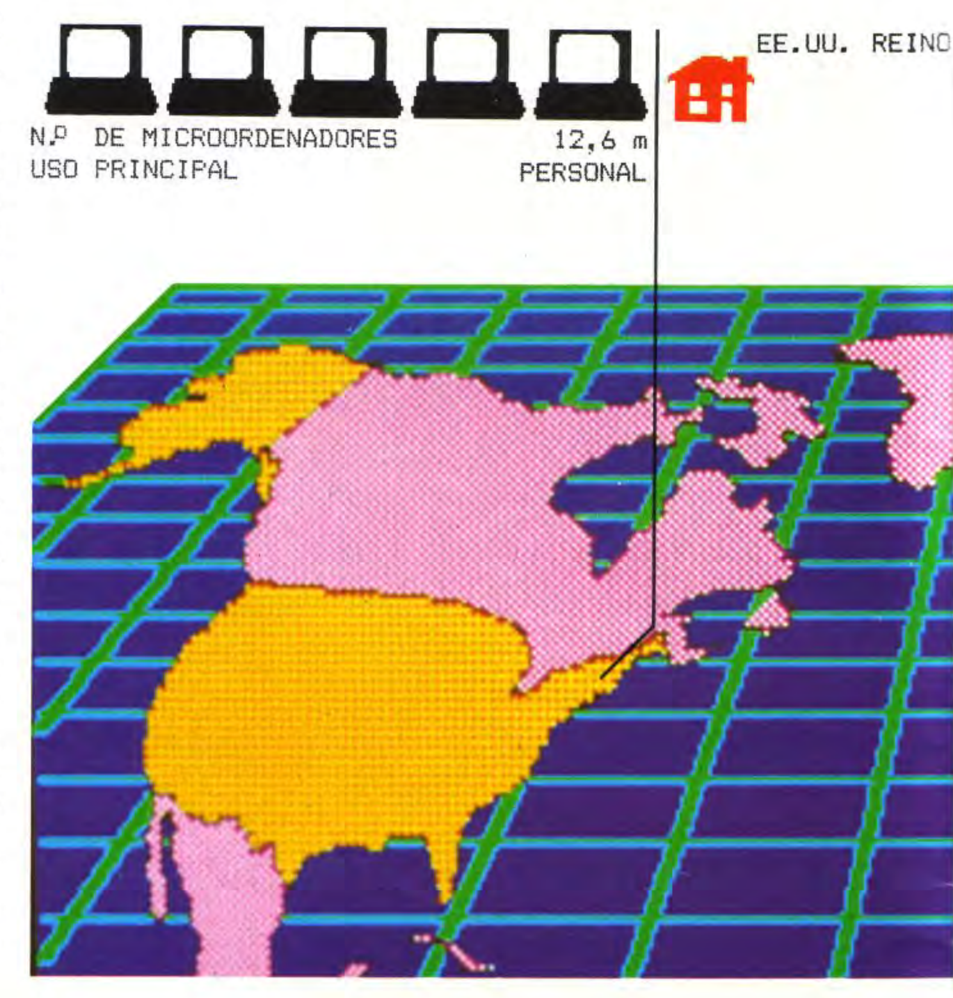
Actividades tan tradicionales como las de tipógrafo y cajista, e incluso la de maestro, podrían ser sustituidas con la actual tecnología. Y dentro de poco acudir a la consulta del médico de cabecera podrá consistir en una entrevista con un terminal de ordenador.

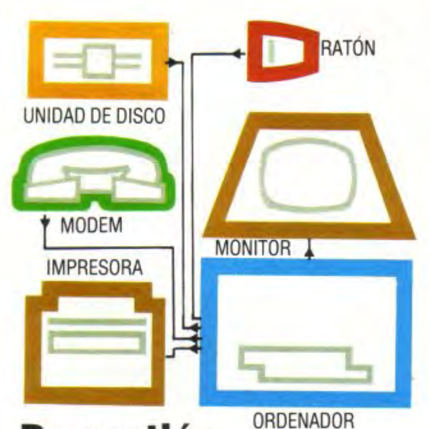
La consecuencia social de la primera revolución industrial fue el desplazamiento de millones de personas desde las zonas rurales y la aparición de la polución industrial (y los beneficios materiales) que caracterizan al mundo occidental. Nos hallamos al borde de una revolución tan dramática como aquélla, en la que el motor de la sociedad será el ordenador. Una revolución en la que la clave para la supervivencia estará en comprender al ordenador y saber utilizarlo.



Personal
Un sistema personal económico puede montarse por poco dinero. Es su pasaporte para el futuro

Juegos
Bastan unos pocos accesorios para acceder a su mundo. Los juegos no sólo entretienen, sino que enseñan los principios básicos de la informática



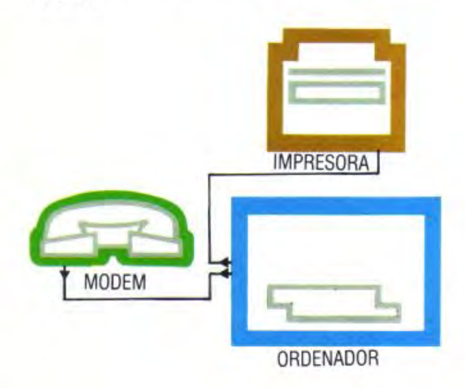


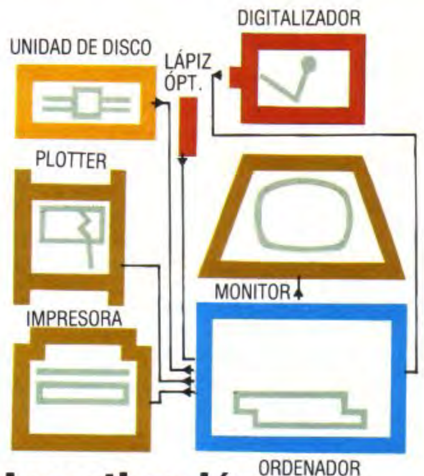
De gestión

Los pequeños sistemas de microordenador están ayudando (y reemplazando) a los empleados administrativos y a los hombres de negocios. Los modems hacen posible el acceso, a través de un teléfono convencional, a ordenadores grandes y a la imprescindible información de sus enormes bancos de datos. Los equipos de mecanógrafos se van convirtiendo en algo ya anticuado, a medida que el tratamiento de textos va ganando terreno. El software contable y por "hojas electrónicas" reduce los costos de la contabilidad y permite que las decisiones financieras se adopten en cuestión de minutos. Las unidades de disco sustituyen salas enteras repletas de archivadores, y la impresora se encarga de todo, desde imprimir cartas perfectas hasta proporcionar una tabulación instantánea de los coeficientes del día en la bolsa de valores. Y el empresario que no sepa digitar será auxiliado por el «ratón», dispositivo similar al mando de bola que permite introducir las órdenes en la máquina

Comunicación

La red de teléfonos es internacional y ahora muchos ordenadores son tan pequeños que pueden sostenerse en la mano o en el regazo. Un atareado ejecutivo puede digitar sus informes a bordo de un Boeing 747 que lo lleva a Nueva York y, a su llegada, enviarlos a casa a la velocidad del electrón. El modem comunica el ordenador personal con el teléfono y permite en teoría un acceso instantáneo a todos los ordenadores del mundo, incluyendo a los ordenadores industriales gigantes. Los bancos de datos están a mano las 24 horas del día y hasta un periodista, que esté consultando el archivo desde Afganistán o Sudamérica, puede redactar su artículo en el tiempo que tarde en hallar una cabina telefónica





Investigación

El potencial de un ordenador avanzado ya ha dejado de ser patrimonio exclusivo del laboratorio de una universidad o de una multinacional. En las escuelas y en los hogares de todo el mundo se están introduciendo sofisticados sistemas. Incluso es posible investigar por CAD (Computer Aided Design, diseño auxiliado por ordenador) y AI (Artificial Intelligence, inteligencia artificial) con un sistema personal. También son asequibles los "plotters" (dispositivos trazadores de gráficos), para realizar dibujos técnicos, y los digitalizadores, que permiten dar entrada a imágenes complejas, como pueden ser mapas o diagramas. En la actualidad, los conocimientos en materia de ordenadores son tan importantes para un currículum de estudios como lo fuera el latín hasta hace algunos años. Los niños del presente están aprendiendo a enfrentarse con el mundo informatizado del mañana

UNIDO
2,2 m
PERSONAL

FRANCIA
0,5 m
DE GESTION

R.F. ALEMANIA
0,6 m
JUEGOS

JAPON
1,1 m
DE GESTION

UNION SOVIETICA Y CHINA:
CIFRAS NO DISPONIBLES

El artista electrónico

Sofisticadas o sencillas, las imágenes por ordenador se componen de miles de puntos diminutos. Y el ordenador debe recordar el color y el brillo de cada uno







Pintando con un micro

Pintar imágenes con un microordenador es sencillo si se dispone de un software para gráficos. Estos programas permiten dibujar complicadas imágenes a través de las órdenes que se digitan en el teclado. Por lo general se empieza colocando líneas rectas en la pantalla. Pueden agregarse círculos, triángulos, cuadrados y otras formas predefinidas. El color puede utilizarse tanto para las líneas como para zonas completas de la pantalla. Las zonas pueden pintarse seleccionando un color, ya sea de una paleta visualizada en pantalla o a través del teclado. La zona que queda encerrada entre sus líneas es coloreada de forma automática. En algunos programas se puede mover un "pincel" a través de la pantalla, como un cursor. La fotografía grande muestra el resultado de este proceso de "construcción"



Usted está pilotando un aeroplano en vuelo rasante sobre las azoteas de los edificios; se le ha detenido un motor; enfrente de usted se dibuja la silueta amenazante de un rascacielos... y en medio de la pista en la que pretendía desesperadamente aterrizar está ardiendo otro avión. Todo esto sucede ante sus ojos, en la pantalla del televisor. Es una forma de gráficos por ordenador.

En el caso de los programas educativos, como los destinados a que los niños aprendan a deletrear o a realizar operaciones aritméticas, es evidente que, a menos que un programa sea visualmente atractivo, no podrá mantener la atención de los pequeños durante mucho tiempo.

Es probable que quienes utilizan los microordenadores con fines empresariales traten básicamente con números que representen las cantidades de dinero que se han recibido o se han gastado, el stock de un artículo determinado, etc. Pero esta clase de información se comprende y se interpreta con más facilidad cuando se muestra en forma de imágenes. La capacidad del ordenador para "volver a dibujar" rápidamente una pintura, para incorporar nueva información, distintas alternativas, etc., y para emitir textos (impresos) cuando sea requerido en ese sentido, resulta asimismo de gran valor para algunas aplicaciones de gestión.

El lienzo del ordenador

¿Cómo crea pinturas el ordenador? Para responder a esta pregunta ocupémonos en primer lugar del "lienzo" sobre el que pinta el ordenador. Un microordenador produce imágenes en su pantalla de visualización de datos iluminando o "encendiendo" uno o varios puntos en una de diversas posiciones en la pantalla. Estos puntos están dispuestos en líneas a través de la pantalla y en columnas a lo largo de ella, de manera que la situación de cada punto viene dada por el lugar que ocupe en línea y columna. La iluminación de ciertos puntos, mientras los restantes permanecen apagados, da como resultado imágenes específicas. Esto es válido no sólo para las pantallas monocromáticas, sino también para las visualizaciones a color. En este caso las pinturas se forman al hacer que los puntos adquieran los colores adecuados.

Para visualizar una simple letra o un número, el ordenador se vale de una matriz de puntos rectangular. Esta matriz se conoce como "matriz de puntos". En un microordenador típico ésta consistirá probablemente en un bloque de ocho líneas que contengan ocho puntos cada una.

El número de puntos en la pantalla no es el mismo para todos los ordenadores, pero una cuadrícula bastante típica estaría compuesta por 192 líneas de 256 puntos cada una; o sea, 192 líneas y 256 columnas.

Es evidente que cuantos más puntos haya en la pantalla de visualización de datos del ordenador, mayor será la definición de las imágenes. El grado de definición observable al visualizar gráficos se denomina *resolución*. Se dice que un ordenador que puede visualizar 192 líneas de 256 puntos cada una, tiene una resolución de 256 × 192. Cuanto más alta sea la resolución —es decir, cuantos más puntos puedan colocarse

en pantalla— menos "granulosa" será la imagen.

Todos los microordenadores poseen una densidad máxima de puntos que pueden visualizarse en sus pantallas para gráficos, pero algunos ejemplares también pueden programarse de modo que utilicen matrices de puntos menos densas. Por ejemplo, un microordenador puede poseer una resolución máxima de 640 × 256; o sea, que le es posible alcanzar una densidad máxima de puntos dada por 256 líneas, cada una de las cuales contiene 640 puntos. Sin embargo, también puede programarse para que utilice sólo 320 de estas columnas, es decir, una resolución de 320 × 256, o incluso resoluciones inferiores si así se le requiere. En una máquina con capacidad para este tipo de variación, la resolución debe determinarse al comienzo de la parte del programa que produce los gráficos.

Las líneas y las curvas producidas por un sistema de gráficos por ordenador basado en puntos en realidad no son continuas, como lo serían trazadas con un lápiz. Son senderos de puntos iluminados que están más o menos próximos entre sí según la resolución del sistema. Un sistema de baja resolución sólo será capaz de producir curvas toscas, y una línea recta tampoco será perfectamente recta, ya que los puntos en la pantalla forman líneas rectas sólo en ciertas direcciones (a lo largo de líneas y columnas y a través de las diagonales formadas por la matriz de puntos). Para trazar una línea recta el sistema de gráficos debe iluminar los puntos más próximos al sendero de la línea indicada. El resultado puede ser un efecto "en escalera". También en este caso, cuanto más alta sea la resolución del sistema, menos evidentes resultarán estas "escaleras".

Como los gráficos se exhiben en una pantalla de televisión, para proporcionar una continuidad de visualización la imagen debe ser continuamente "refrescada" o "redibujada", porque, de lo contrario, sólo aparecería por un instante y luego desaparecería. Por este motivo, la imagen debe representarse de alguna manera en el ordenador para que, en caso necesario, éste pueda tomarla como referencia. En realidad la representación de la imagen se almacena en una zona especial de la memoria del ordenador denominada *memoria de pantalla*. En una visualización monocromática, cada punto de la pantalla corresponde a un bit de la memoria de pantalla. Una imagen se representa estableciendo que los bits que corresponden a los puntos iluminados sean 1 y dejando 0 para los bits correspondientes a los puntos no iluminados. De esta manera, si un ordenador puede mantener una visualización monocromática con una resolución de 256 × 192, debe poseer una memoria de pantalla de 256 × 192 o 49 152 bits, es decir, 6 kilobytes, ya que 1 kilobyte es igual a 8 192 bits.

Para gráficos a color se necesita más memoria. Dos bits pueden representar cuatro colores diferentes, como en el ejemplo siguiente:

bit 1	bit 2	*color*
0	0	blanco
0	1	rojo
1	0	azul
1	1	negro

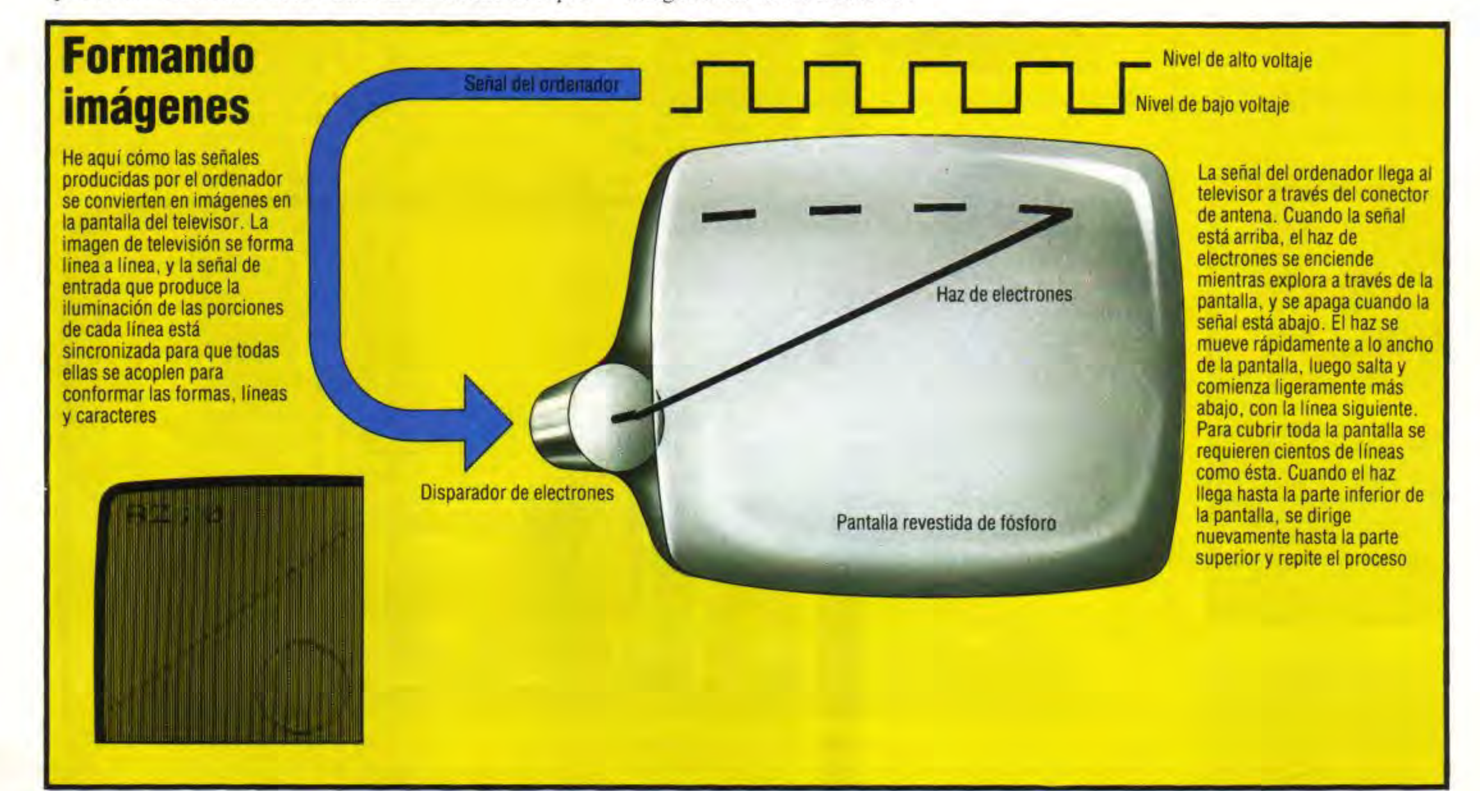
Para representar cualquier imagen en cuatro colores, es preciso asignarle dos bits a cada punto de la pantalla. Del mismo modo, para imágenes en ocho colores, a cada punto de la pantalla deben asignársele tres bits, mientras que para visualizaciones en 16 colores corresponderían cuatro bits para cada punto, y así sucesivamente. Por tanto, un microordenador con capacidad para visualizar colores con una resolución de 160 × 256, debe tener una memoria de pantalla de 160 × 256 × 4, que equivale a 160 kilobits (20 kilobytes).

La necesidad de la memoria de pantalla explica, en parte, por qué algunos ordenadores están diseñados para funcionar a distintas resoluciones. Si no posee suficiente memoria de pantalla, el ordenador no puede almacenar y, en consecuencia, no puede visualizar imágenes de alta resolución.



Creando gráficos

Producir imágenes con el ordenador personal puede ser sencillo. La mayoría de los modelos cuentan con órdenes especiales en BASIC para ayudar a "dibujar" o definir en la pantalla tanto a un "invasor extraterrestre" como a un auténtico cuadro. Asimismo, existe un software especial para crear dibujos animados en la pantalla.



Formando imágenes

He aquí cómo las señales producidas por el ordenador se convierten en imágenes en la pantalla del televisor. La imagen de televisión se forma línea a línea, y la señal de entrada que produce la iluminación de las porciones de cada línea está sincronizada para que todas ellas se acoplen para conformar las formas, líneas y caracteres

La señal del ordenador llega al televisor a través del conector de antena. Cuando la señal está arriba, el haz de electrones se enciende mientras explora a través de la pantalla, y se apaga cuando la señal está abajo. El haz se mueve rápidamente a lo ancho de la pantalla, luego salta y comienza ligeramente más abajo, con la línea siguiente. Para cubrir toda la pantalla se requieren cientos de líneas como ésta. Cuando el haz llega hasta la parte inferior de la pantalla, se dirige nuevamente hasta la parte superior y repite el proceso

Bits y bytes

El ordenador sólo comprende números. Pero son números de apariencia un tanto extraña

Siempre que se escribe acerca de ordenadores, se utilizan las palabras *bits* y *bytes*. Estos vocablos describen la manera en que los ordenadores almacenan y emplean los números.

Y lo hacen de un modo bastante diferente al nuestro. Nosotros representamos los números mediante 10 símbolos diferentes (de 0 a 9) y los manejamos en múltiplos de 10. (Esto se conoce como una "base" de 10.) Los ordenadores, por el contrario, para llevar a cabo toda su magia matemática se sirven sólo de dos números: el cero y el uno. Los bits y los bytes son formas de representar combinaciones entre estos dos números.

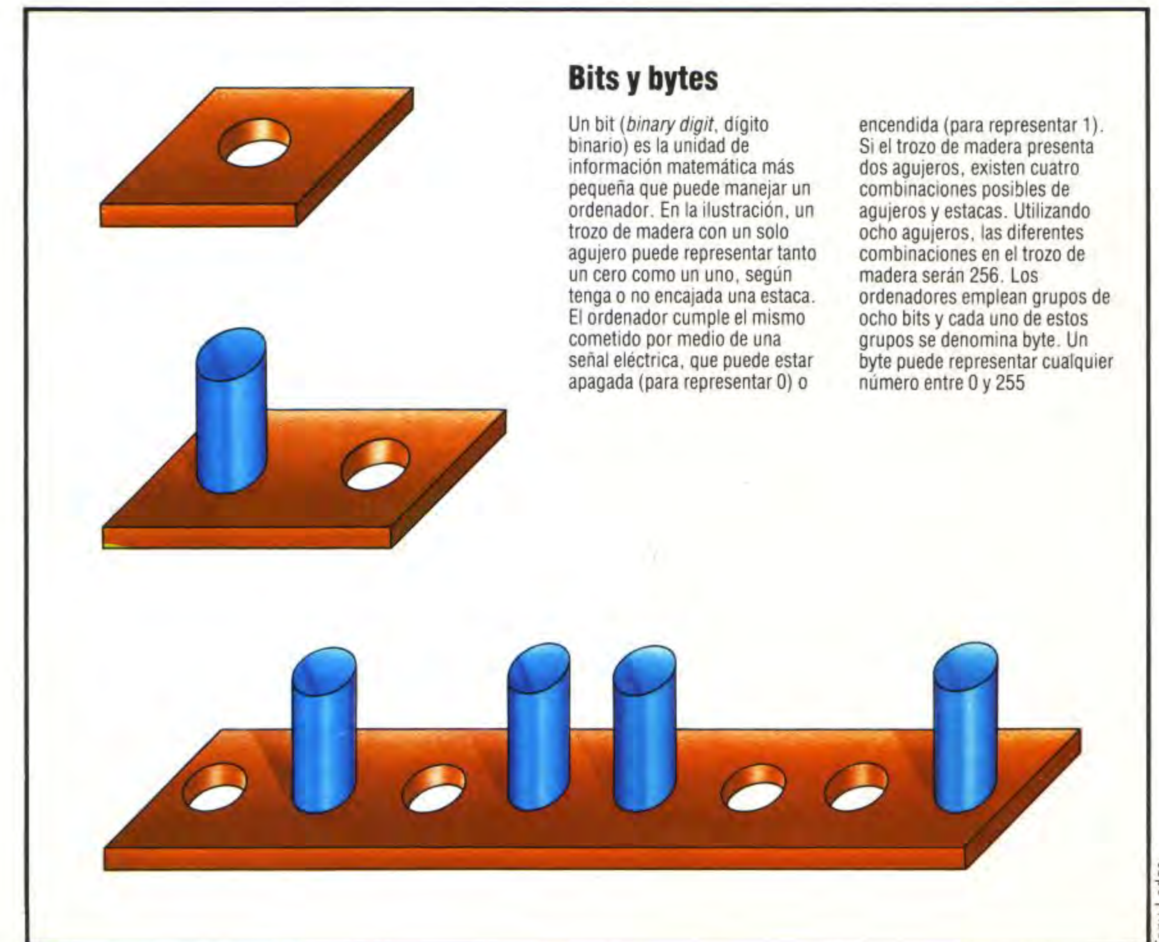
Un bit es la unidad de información más pequeña que puede manejar un ordenador. Es la forma que tiene el ordenador de representar los números cero y uno. Un grupo de ocho bits se denomina byte; un byte permite al ordenador representar cifras muy elevadas.

Ocupémonos en primer lugar de los bits: qué son y por qué se denominan así. Los ordenadores son dispositivos electrónicos y, en consecuencia, todas sus funciones las realizan, en última instancia, con señales eléctricas. Una señal eléctrica puede estar "encendida" o "apagada"; éste es el principio por el cual las señales eléctricas pueden representar números.

La ilustración muestra un trozo de madera que presenta un agujero en el cual puede encajarse una estaca. Aun cuando encaje en un solo agujero, la estaca puede representar dos números y constituye una analogía excelente para describir la forma en que funciona un ordenador. El agujero puede no estar ocupado por una estaca, en cuyo caso representa un cero, o bien estar ocupado por ella, en cuyo caso representa un uno. El mismo trozo de madera puede, por tanto, simbolizar tanto un cero como un uno.

En el ordenador se consigue el mismo efecto con una señal eléctrica: si está apagada representa un cero; si está encendida, un uno. Un cable, o un trozo de madera con un agujero, puede, pues, utilizarse para representar dos condiciones: sin estaca o con estaca, ausente o presente, encendido o apagado, 0 o 1.

Esta unidad mínima de información se denomina *bit*. La palabra bit denota su reducido tamaño y representa dos posibles situaciones; deriva de **BI***nary* *digi***T**. Considerado desde otra perspectiva, un bit



Bits y bytes

Un bit (*binary digit*, dígito binario) es la unidad de información matemática más pequeña que puede manejar un ordenador. En la ilustración, un trozo de madera con un solo agujero puede representar tanto un cero como un uno, según tenga o no encajada una estaca. El ordenador cumple el mismo cometido por medio de una señal eléctrica, que puede estar apagada (para representar 0) o encendida (para representar 1). Si el trozo de madera presenta dos agujeros, existen cuatro combinaciones posibles de agujeros y estacas. Utilizando ocho agujeros, las diferentes combinaciones en el trozo de madera serán 256. Los ordenadores emplean grupos de ocho bits y cada uno de estos grupos se denomina byte. Un byte puede representar cualquier número entre 0 y 255

Tony Lodge



0	00000000	128	10000000
1	00000001	129	10000001
2	00000010	130	10000010
3	00000011	131	10000011
4	00000100	132	10000100
5	00000101	133	10000101
6	00000110	134	10000110
7	00000111	135	10000111
8	00001000	136	10001000
9	00001001	137	10001001
10	00001010	138	10001010
11	00001011	139	10001011
12	00001100	140	10001100
13	00001101	141	10001101
14	00001110	142	10001110
15	00001111	143	10001111
16	00010000	144	10010000
17	00010001	145	10010001
18	00010010	146	10010010
19	00010011	147	10010011
20	00010100	148	10010100
21	00010101	149	10010101
22	00010110	150	10010110
23	00010111	151	10010111
24	00011000	152	10011000
25	00011001	153	10011001
26	00011010	154	10011010
27	00011011	155	10011011
28	00011100	156	10011100
29	00011101	157	10011101
30	00011110	158	10011110
31	00011111	159	10011111
32	00100000	160	10100000
33	00100001	161	10100001
34	00100010	162	10100010
35	00100011	163	10100011
36	00100100	164	10100100
37	00100101	165	10100101
38	00100110	166	10100110
39	00100111	167	10100111
40	00101000	168	10101000
41	00101001	169	10101001
42	00101010	170	10101010
43	00101011	171	10101011
44	00101100	172	10101100
45	00101101	173	10101101
46	00101110	174	10101110
47	00101111	175	10101111
48	00110000	176	10110000
49	00110001	177	10110001
50	00110010	178	10110010
51	00110011	179	10110011
52	00110100	180	10110100
53	00110101	181	10110101
54	00110110	182	10110110
55	00110111	183	10110111
56	00111000	184	10111000
57	00111001	185	10111001
58	00111010	186	10111010
59	00111011	187	10111011
60	00111100	188	10111100
61	00111101	189	10111101
62	00111110	190	10111110
63	00111111	191	10111111
64	01000000	192	11000000
65	01000001	193	11000001
66	01000010	194	11000010
67	01000011	195	11000011
68	01000100	196	11000100
69	01000101	197	11000101
70	01000110	198	11000110
71	01000111	199	11000111
72	01001000	200	11001000
73	01001001	201	11001001
74	01001010	202	11001010
75	01001011	203	11001011
76	01001100	204	11001100
77	01001101	205	11001101
78	01001110	206	11001110
79	01001111	207	11001111
80	01010000	208	11010000
81	01010001	209	11010001
82	01010010	210	11010010
83	01010011	211	11010011
84	01010100	212	11010100
85	01010101	213	11010101
86	01010110	214	11010110
87	01010111	215	11010111
88	01011000	216	11011000
89	01011001	217	11011001
90	01011010	218	11011010
91	01011011	219	11011011
92	01011100	220	11011100
93	01011101	221	11011101
94	01011110	222	11011110
95	01011111	223	11011111
96	01100000	224	11100000
97	01100001	225	11100001
98	01100010	226	11100010
99	01100011	227	11100011
100	01100100	228	11100100
101	01100101	229	11100101
102	01100110	230	11100110
103	01100111	231	11100111
104	01101000	232	11101000
105	01101001	233	11101001
106	01101010	234	11101010
107	01101011	235	11101011
108	01101100	236	11101100
109	01101101	237	11101101
110	01101110	238	11101110
111	01101111	239	11101111
112	01110000	240	11110000
113	01110001	241	11110001
114	01110010	242	11110010
115	01110011	243	11110011
116	01110100	244	11110100
117	01110101	245	11110101
118	01110110	246	11110110
119	01110111	247	11110111
120	01111000	248	11111000
121	01111001	249	11111001
122	01111010	250	11111010
123	01111011	251	11111011
124	01111100	252	11111100
125	01111101	253	11111101
126	01111110	254	11111110
127	01111111	255	11111111

Bytes en memoria

Los bytes son grupos de ocho dígitos binarios (bits). El ordenador utiliza los bytes para almacenar números entre 0 y 255. Cada byte se almacena en una "celda" de memoria separada y estas celdas están dispuestas sistemáticamente para que el ordenador pueda hallar el byte que necesite. Para ello es necesario que el ordenador conozca la localización de la caja de memoria que contiene el byte

1.ª localización de memoria ▶

2.ª localización de memoria ▶

3.ª localización de memoria ▶

Tony Lodge

puede servir para contar, pero sólo de cero a uno.

Un trozo de madera con dos agujeros puede simbolizar cuatro casos diversos, o contar de cero a tres. Ambos agujeros pueden estar vacíos; el agujero derecho puede estar ocupado por una estaca; el agujero izquierdo puede tener una estaca, o bien ambos agujeros pueden estar ocupados. La parte inferior de la ilustración de la página contigua muestra un trozo de madera con ocho agujeros. En este caso existen 256 permutaciones posibles entre agujeros y estacas, y éstas están reflejadas en la tabla: los unos representan a las estacas y los ceros a los agujeros.

Este grupo formado por ocho dígitos binarios (bits) se denomina *byte*. Un byte puede, pues, representar 256 situaciones diferentes (y contar de 0 hasta 255).

Cuando decimos que un ordenador "almacena" un byte, significa que su memoria conserva un número (entre 0 y 255) que utilizará cuando sea requerido en ese sentido. Cada byte posee su propia "caja" (dirección de memoria) y estas "cajas" se disponen en un orden preestablecido (el gráfico superior ilustra cómo están apiladas una encima de la otra). Cuando el ordenador necesita recuperar un número, sólo ha de saber en qué caja (dirección) está almacenado el byte.

Todos los números entre 0 y 255 pueden representarse mediante singulares combinaciones de unos y ceros (tabla a la izquierda). Los bits son almacenados y utilizados por el ordenador en grupos de ocho. Cada grupo de ocho bits se denomina byte

Oric-1

Este modelo puede crear magníficos gráficos en color y una amplia gama de efectos sonoros

El Oric-1, un pequeño ordenador de fabricación británica, compite con el Sinclair Spectrum tanto en precio como en prestaciones. Posee una atractiva carcasa plástica de color gris y su teclado, con un ángulo de inclinación que resulta muy cómodo para digitar, tiene teclas móviles individuales y permite la escritura al tacto.

Existen dos versiones disponibles; la más cara ofrece 48 Kbytes, suficientes para almacenar complejos programas.

El Oric posee las conexiones habituales para televisor, cassette y otras unidades. Se puede conectar con una impresora y dispone de otro conector para enchufar memoria extra, cartuchos de programa y un modem.

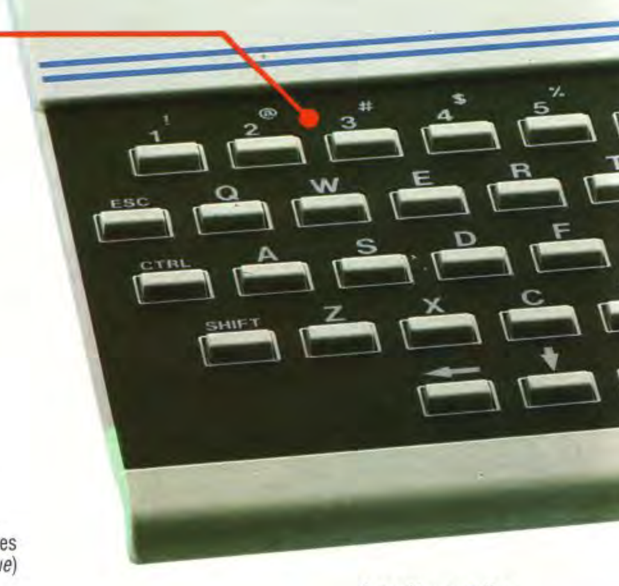
El modem es un accesorio particularmente atractivo. Permite que el Oric pueda comunicarse con otros ordenadores a través del teléfono.

El Oric tiene BASIC incorporado y puede, asimismo, funcionar con otros lenguajes. La versión de 48 K se suministra con FORTH y con BASIC.

Utilizando su BASIC incorporado, el Oric puede producir sonido y gráficos en color. También puede visualizar ocho colores y crear y almacenar caracteres de cualquier forma. Mediante las órdenes PAPER (papel) e INK (tinta), es posible cambiar el color de cualquiera de estos caracteres "definidos" y el color del fondo sobre el cual están fijados.

El sonido del Oric es tan impresionante como sus gráficos. Unas órdenes especiales permiten crear una gran variedad de sonidos y música, con notas y acordes que pueden abarcar seis octavas.

El Oric es un micro barato de gran versatilidad, especialmente indicado para los principiantes y para juegos. Su potencial de expansión lo hace especialmente atractivo y su capacidad de comunicarse a través del teléfono representa un reclamo adicional.



Teclado
El teclado del Oric posee 57 teclas móviles. La disposición de las teclas de letras y números responde al trazado QWERTY estándar. Las teclas ESCAPE y CONTROL están a la izquierda, y DELETE y RETURN a la derecha. La línea inferior del teclado contiene las teclas espaciadoras y de control del cursor. Como la disposición de las teclas es similar a la de una máquina de escribir y se accionan individualmente, con un poco de práctica se puede digitar en este teclado a considerable velocidad

Conexión RGB
Permite conectar al Oric con monitores en color utilizando señales separadas rojas, verdes y azules (RGB: *red, green, blue*) para una visualización en pantalla de mejor calidad

Conexión cassette
La grabadora de cassette se conecta al ordenador por medio de este enchufe

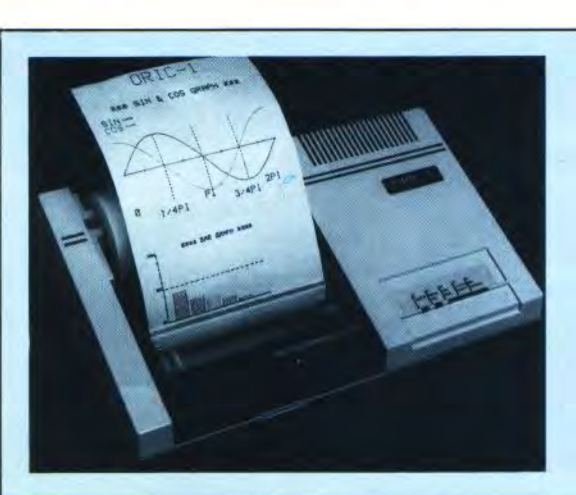
Modulador RF
La señal de video producida por el ordenador no puede alimentarse directamente a la conexión de antena de televisión. Este circuito convierte la señal para adecuarla a un televisor normal

Conexión televisión
A través de esta conexión el Oric se acopla a un aparato de televisión

Reloj
El reloj electrónico marca el compás un millón de veces por segundo para regular y sincronizar todas las operaciones que ejecuta el Oric



RAM
Estos chips constituyen la memoria central del Oric; retienen el software mientras el ordenador está en funcionamiento. La RAM "olvida" su contenido apenas se interrumpe la alimentación eléctrica



La impresora
El diseño de la impresora en color armoniza con el Oric-1. Puede imprimir textos y trazar gráficos en cuatro colores: rojo, verde, negro y azul.
La impresora utiliza cuatro plumas esferográficas, una para cada color, que escriben sobre un rollo de papel de 11 cm de ancho. Por lo que respecta a la escritura de textos, la realiza con caracteres de 15 tamaños distintos y en cuatro ángulos diferentes. Con este grado de flexibilidad, la impresora, que se conecta mediante interface paralela tipo Centronics Standard, puede efectuar otros muchos cometidos



Interface para impresora
Este enchufe proporciona una conexión estándar a través de la cual se puede acoplar al ordenador una impresora

Puerta de expansión
Esta conexión permite adaptar al Oric una gran variedad de dispositivos, incluyendo cartuchos de programas y equipo de comunicaciones

Enchufe red
Aquí se conecta la fuente de alimentación eléctrica proveniente del transformador propio del Oric

Disipador
Las altas temperaturas pueden dañar con facilidad los circuitos electrónicos. Esta placa disipa todo el calor excesivo que se genera

ROM
La memoria ROM almacena software permanentemente. La función principal de los chips ROM es el almacenamiento del lenguaje de programación BASIC

Microprocesador
Este chip es el caballo de tiro que proporciona al ordenador la capacidad de realizar cálculos. Se trata de un microprocesador 6502A de la Rockwell

Entrada/Salida
Este chip convierte las señales que entran en el Oric de forma que su CPU pueda utilizarlas. La información de salida es convertida en la clase de señal que requieran la impresora y los otros dispositivos

Chip de sonido
Un chip diseñado especialmente para producir notables efectos sonoros

Chip video
El Oric es muy apreciado por su buena resolución en color. El chip genera una señal que contiene información en color. La señal llega hasta la pantalla del televisor a través del modulador

Altavoz
La calidad de sonido es sorprendente. El Oric utiliza su propio altavoz, no el del televisor

ORIC-1

DIMENSIONES
280×178×150 mm

PESO
848 g

CPU
6502A

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
16 Kbytes de RAM ampliables hasta 48 Kbytes.
16 Kbytes de ROM con BASIC incorporado

VISUALIZACION EN VIDEO
4 modalidades: una modalidad para textos con 28 líneas de 40 caracteres y dos modalidades para gráficos de baja resolución en la misma cuadrícula; modalidad para alta resolución, con 200×240 puntos

INTERFACES
Conexión TV, conexión cassette y alta fidelidad, interface para impresora Centronics, video RGB, conexión de expansión

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
FORTH

VIENE CON
Unidad de alimentación eléctrica con enchufe integral, conexión TV, conexiones cassette, cassette de juegos, manual

TECLADO
57 teclas móviles individuales, incluyendo barra espaciadora

DOCUMENTACION
Un manual de programación BASIC que describe la máquina y su lenguaje con un estilo muy ágil. Los capítulos relativos a la programación BASIC se refieren a la manipulación de números y palabras y explican cómo utilizar los gráficos en color y los recursos de sonido del Oric para conseguir buenos resultados.
Incluye muchos programas cortos que resultan muy instructivos para un primer nivel. Los apéndices indican la manera de organizar la información; no tiene índice. En líneas generales, la documentación es muy adecuada para un principiante

Toma de contacto

Al principio parece que todos los teclados son iguales; sin embargo, unos son mejores que otros y trabajan de forma diferente

El teclado de un ordenador constituye una parte importante del sistema del mismo. Después de todo, es el medio por el que usted puede comunicarse con el ordenador. La importancia del teclado es equiparable a la de la capacidad de memoria, o a la calidad de los gráficos.

En los microordenadores se ha adoptado el teclado tipo QWERTY, similar a los de las máquinas de escribir; este tipo de teclado recibe dicha denominación porque las seis primeras letras de la línea superior forman la palabra QWERTY.

A comienzos de la década de los cincuenta, cuando los ordenadores comenzaron a utilizarse con fines comerciales, el diseño QWERTY, el sistema de mecanografía convencional, se convirtió en el dispositivo de entrada estándar para los ordenadores. En la actualidad el propietario de un ordenador debe someterse al QWERTY, un sistema muy fácil para un mecanógrafo experimentado, pero que para el recién iniciado suele resultar difícil.

Cuando el precio de los ordenadores ascendía a cifras muy altas, el coste de un teclado mecánico era insignificante. Sin embargo, los sucesivos desarrollos en la tecnología de los microprocesadores redujeron notablemente el coste de los componentes electrónicos de los mismos.

Cuando apareció el Sinclair ZX81, un teclado del tipo de una máquina de escribir podía representar una parte significativa del coste de fabricación de un microordenador. El teclado móvil mecánico, en el que se basa el Dragon, por ejemplo, tiene auténticos interruptores debajo de las teclas (véase la ilustración de la página siguiente). Cuando se pulsa la tecla, los contactos internos se cierran para completar un circuito. Los interruptores de este tipo contienen numerosos componentes y aumentan considerablemente el coste del teclado.

La solución a este problema la constituye un nuevo tipo de teclado, más económico. El teclado "sensible al tacto" del ZX81 se desarrolló a partir de la idea de que la mayoría de las personas que adquieran un microordenador estarían interesadas principalmente en utilizarlo para juegos y para escribir pequeños programas.

Estas aplicaciones implican una actividad mínima con el teclado, por lo cual parecía lógico que los posibles usuarios de un microordenador estuviesen preparados para decidirse por un teclado de inferior calidad. Aunque se prescinda de las ventajas de un teclado convencional del tipo de máquina de escribir, se obtiene, en cambio, un ahorro considerable.

El ZX81 fue diseñado con un teclado sensible al tacto, eliminando la mayoría de las piezas. Este recurso propició el abaratamiento del modelo, pero no representó una solución definitiva. El teclado sensible al tacto ofrece el inconveniente de que no proporciona mucha "realimentación táctil" (o sea, usted nunca está seguro de que la tecla pulsada ha sido registrada en el ordenador, a menos que lo compruebe mirando la pantalla).

En su siguiente producto (el Spectrum), la Sinclair introdujo el teclado de membrana (véase diagrama). Este tipo de teclado representa una nueva mejora, pero todavía carece de la realimentación táctil de que goza el teclado de tipo máquina de escribir.

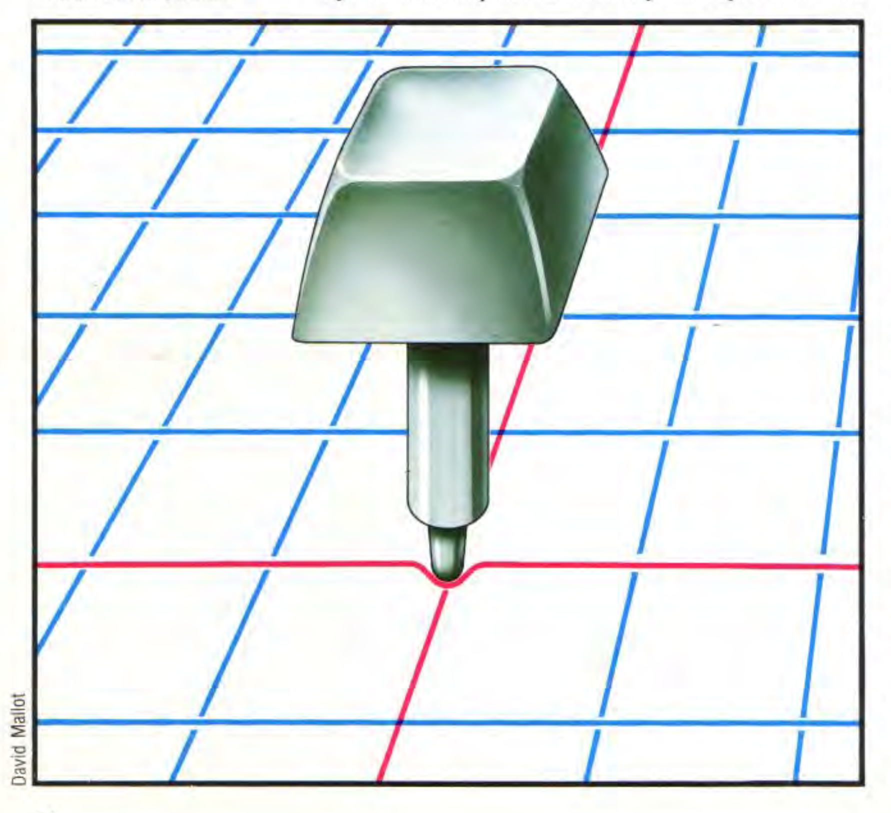
Algunos ordenadores relativamente baratos (entre los que se incluye el Dragon) poseen teclados de máquina de escribir "profesional". Las ventajas de los teclados de este tipo se ponen claramente de manifiesto cuando el ordenador es sometido a una intensa utilización para tratamiento de textos. La familiar sensación de que se está trabajando con una máquina de escribir permite que la tarea se efectúe con muchísima rapidez.

Entre el teclado completamente móvil y el de tipo membrana del Spectrum, existe otro tipo de teclados, que a menudo son denominados "teclados similares al de las máquinas de calcular" (por ejemplo, los que incorpora el New Brain y el Oric-1). Las teclas proporcionan una "sensación" mejor, pero son pequeñas, rígidas y resultan menos aptas para la escritura al tacto que las teclas completamente móviles del tipo de máquina de escribir.

Una forma de superar parcialmente la falta de realimentación táctil de los teclados sensibles al tacto y de

La matriz del teclado

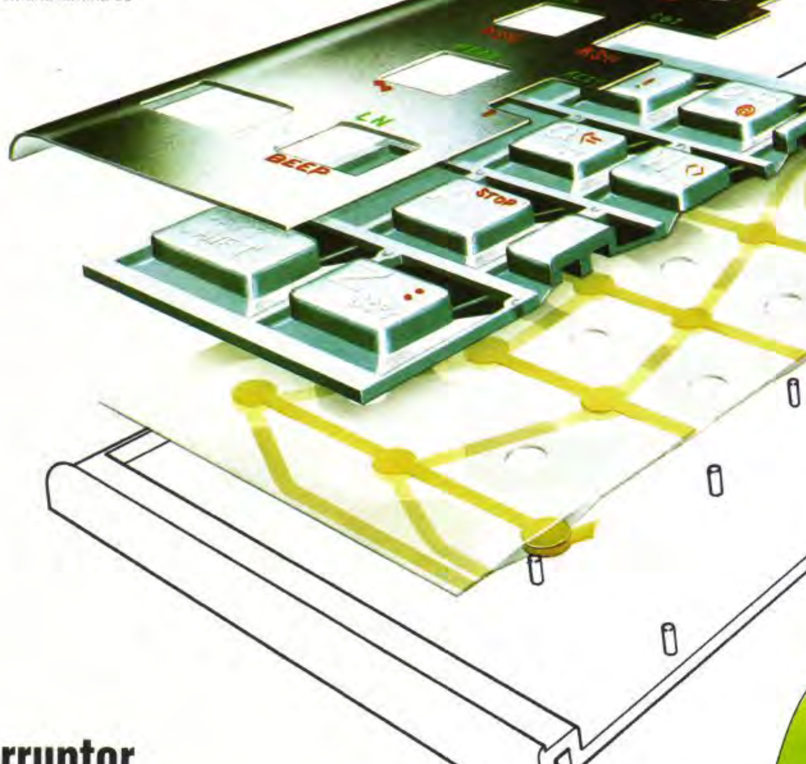
Las teclas de un ordenador son, en realidad, interruptores que están conectados a una rejilla de cables. En la ilustración vemos cómo, cuando es pulsada una tecla, se conectan dos de los cables de la rejilla. A cada tecla corresponde únicamente un par de cables, por lo que sólo efectúa una conexión en la rejilla, permitiendo que el ordenador descubra cuál es la tecla que ha sido pulsada



David Mallot

El teclado Sinclair

La Sinclair revolucionó el arte del diseño de teclados con el denominado de tipo "membrana" del Spectrum, que consiste en una lámina de caucho moldeado, con protuberancias en forma de teclas, montada sobre un relleno de contactos que conforman la matriz o la rejilla del teclado. Cuando se pulsa una de estas teclas, una protuberancia situada debajo de la misma hace que los contactos se unan. El ordenador verifica qué contactos se han cerrado y puede deducir a qué tecla corresponden. Los contactos cerrados por la tecla normalmente se mantienen separados mediante una burbuja de aire atrapada entre dos láminas de plástico. La fuerza que hace saltar las teclas devolviéndolas a su posición inicial proviene de la propia elasticidad del caucho, que se estira cuando la tecla es pulsada. Este enfoque original a la ingeniería de teclados ciertamente ha eliminado algunos elementos, pero significa un abaratamiento en los costes de fabricación del producto, lo que redunda en beneficio del consumidor



Interruptor de tecla

Los interruptores de teclas del tipo de máquinas de escribir incorporan, por lo general, un par de contactos. Éstos normalmente están separados y no permiten que pase corriente eléctrica. Cuando se pulsa la tecla, una pieza de plástico moldeado (en color malva en la ilustración) se mueve hacia abajo y permite que los contactos se acerquen y cierren un circuito. Un muelle interior hace que la tecla vuelva a su posición. Cuando se cierran los contactos se produce un flujo de corriente que es detectado por el ordenador. Los cables conectados a los contactos de cada interruptor están dispuestos en una rejilla. Para saber cuál es la tecla que ha sido pulsada, el ordenador verifica cuáles son los cables "verticales" de la rejilla y cuáles los "horizontales" que conducen corriente. Las teclas de este tipo son muy complejas desde el punto de vista mecánico y los costes de fabricación son más elevados. Ofrecen una gran fiabilidad y proporcionan una sensación más "positiva" que las teclas de membrana plástica. Esta sensación táctil proviene de la resistencia que ofrece el muelle. Una tecla bien diseñada proporciona tal realimentación táctil que el usuario sabe si ha pulsado la tecla de forma correcta. Las cabezas de las teclas también están modeladas para que la digitación resulte más cómoda. Los teclados de este tipo son los más indicados si el ordenador ha de utilizarse de forma intensiva

David Weeks

los del tipo membrana, consiste en emitir un "beep" (sonido corto y agudo) cada vez que se pulsa una tecla. Mediante dicha señal sonora, el usuario puede advertir que la tecla ha sido pulsada y reconocida por el ordenador.

Los diseñadores del Sinclair ZX81 y del Spectrum introdujeron una forma inédita y muy útil de reducir la cantidad de pulsaciones mecanográficas que se precisan para la entrada de programas BASIC. Cada tecla representa algo más que una letra del alfabeto o un simple número. Mediante la utilización de una tecla de "función" especial junto a las teclas normales, puede conseguirse que aparezcan en pantalla palabras completas en BASIC sin necesidad de digitarlas letra por letra. Por ejemplo, la palabra PRINT en BASIC puede ser elaborada simplemente pulsando la tecla de función especial y la tecla de la letra P simultáneamente. En su modelo M5, la firma Sord desarrolla una idea similar.

El fin de la digitación

Hasta no hace mucho tiempo, el único modo de lograr que un ordenador realizara alguna operación consistía en introducir en él las instrucciones digitándolas en el teclado. Esto, con frecuencia, resultaba un trabajo tedioso, que empeoraba más si cabe en el caso de que el usuario no fuera un mecanógrafo diestro y veloz. Comprendiendo que para muchas personas esta dificultad suponía una barrera que las disuadía de utilizar (y, desde luego, de adquirir) los ordenadores, los fabricantes hallaron una solución tan sencilla como brillante: el "ratón" (similar al mando de bola). El "ratón" puede desplazarse a través de cualquier superficie plana y, a medida que lo hace, el cursor se mueve asimismo a través de la pantalla de visualización de datos. De esta manera, uno puede moverse rápidamente hasta cualquier sector de la pantalla, y, pulsando el botón, se inicia la operación que se desea. Los "ratones" también pueden utilizarse para realizar gráficos, trazar líneas o rellenar con colores la pantalla

El educador electrónico

Incluso a los más pequeños de la familia les encantará usar el ordenador. Aquí le explicamos cuál es la mejor forma de iniciarlos

Uno de los instrumentos por ordenador más eficaces para niños a nivel de enseñanza primaria es la "tortuga". El robot está conectado a un microprocesador y funciona según un programa denominado LOGO. Los niños pueden dibujar con la tortuga, que resulta muy útil para enseñar conceptos matemáticos como forma, distancia y la relación entre los objetos. ¡Y además es sumamente divertida!



Ian McKinnell

Muchos padres se preguntan si un ordenador personal podría resultar útil para sus hijos. Gran parte de ellos ya conocen las ventajas que supone para un adolescente aprender el manejo de los ordenadores, tanto en su casa como en la escuela; pero ¿tiene utilidad el uso del ordenador para los niños más pequeños?

¡Sí! La respuesta es decididamente afirmativa, aunque existen diferentes maneras de introducir al niño en el manejo del ordenador, y algunas de ellas mejores que otras.

Muchos de los gobiernos de los países desarrollados, convencidos de la necesidad de que los niños aprendan el manejo de los ordenadores en la escuela, están realizando importantes inversiones para dotar a todos los centros de enseñanza primaria cuando menos de un microordenador. Ahora son los propios maestros quienes deben decidir cuál es la mejor manera de aprovechar este "pequeño potencial" para que resulte útil en sus programas de enseñanza.

Los ordenadores no sólo resultan idóneos para la enseñanza de las matemáticas; con un buen programa, aunque en la actualidad son muy escasos, los ordenadores pueden ayudar al niño a aprender música, ballet, geografía, idiomas y, por supuesto, materias relacionadas con las matemáticas, como aritmética y geometría.

El ordenador puede ayudar a los pequeños de dos formas principales. El niño puede utilizar la máquina para explorar su propio mundo, o bien el ordenador es capaz de actuar como un maestro, instruyendo y adiestrando al niño en una amplia gama de temas educativos.

No constituye una buena idea intentar que su hijo de seis años aprenda a programar un ordenador en BASIC. Antes de los 12 años, un niño no es capaz de comprender los conceptos abstractos de un lenguaje semejante. Pese a que algunos niños están en disposición de escribir programas en BASIC a los nueve años o

incluso antes, los trabajos del psicólogo francés Jean Piaget nos muestran que antes de los 12 o 13 años la mayor parte de los niños tienen dificultades para captar ideas abstractas.

En consideración a este problema, los investigadores han descubierto un sistema para que el niño controle y programe un ordenador sin necesidad de manejar esos conceptos abstractos (véase recuadro relativo al LOGO). La forma habitual en que el maestro introduce al niño en el mundo del ordenador constituye una mezcla de ambos métodos.

El aprendizaje jugando con la tortuga

Incluso los niños más pequeños pueden utilizar el ordenador para aprender. En la ilustración de la página contigua se ve a un niño jugando con la "tortuga", un robot mecánico que está conectado a un microordenador. Estas tortugas resultan caras y están ideadas para su utilización en la escuela, pero su fundamento es muy simple: el ingenio tiene dos ruedas y está provisto de un lápiz. El niño le comunica a la tortuga que avance a través de una hoja de papel y, a medida que se va desplazando, le indica si debe o no trazar una línea (la "huella" de la tortuga). De esta manera el niño dibuja, instruyendo al robot en cómo debe formar esquinas y unir líneas. Como los niños son estimulados para determinar exactamente qué movimientos debe hacer la tortuga al objeto de que resulte un dibujo con formas específicas, llegan a descubrir por sí mismos los elementos que conforman la geometría básica. Este planteamiento de "autoayuda" constituye la base del método LOGO. La teoría consiste en que las lecciones aprendidas "heurísticamente" (mediante pruebas y errores) se comprenden mucho mejor que cuando se muestran ejemplos.

Estas dos escuelas de pensamiento nos señalan dos formas diferentes de utilizar los ordenadores con los niños. De acuerdo con el LOGO, los niños de nueve o diez años comienzan con el manejo de una versión de la tortuga en la pantalla del ordenador, dibujando formas complicadas y enseñando al robot cómo recordar diversos procedimientos. Cuando un niño "instruye" a la tortuga para que realice líneas o figuras, ya sea sobre una hoja de papel o en la pantalla, no hace otra cosa que programar la máquina. El LOGO es un lenguaje mediante el cual el niño programa antes de que haya desarrollado la capacidad de comprensión de los conceptos abstractos necesarios para muchos lenguajes de ordenador. Por lo tanto, a través del "juego con la tortuga" los niños más pequeños se familiarizan con la idea del manejo del ordenador y pueden explorar su propio entorno.

El otro planteamiento se vale de la evidente "paciencia" del ordenador para enseñar al niño mediante ejemplos.

A los niños que tienen dificultad para comprender un tema o una idea se les suele ayudar con programas de "procedimiento correcto y ejercicios", los cuales formulan preguntas al niño y luego, en un marcador, le indican cuántas ha acertado. Muchos de estos programas son enormemente atractivos, ya que están provistos de excelentes gráficos a color e interesantes melodías o efectos sonoros. Estos cuestionarios estimulan al niño a aprender, y el ordenador no se cansa ni abandona mientras el niño insista con una respuesta equivocada. Esta paciencia del ordenador se ha mostrado muy valiosa para enseñar a los alumnos de comprensión lenta, y los programas de ejercicios en los que, por ejemplo, el niño ha de seleccionar un sustantivo de entre un grupo de palabras, o formar una palabra a partir de una serie de letras, resultan herramientas muy útiles para la educación. Sin embargo, la utilización del ordenador de esta manera nos lleva a sustituir al maestro humano, y de ahí surge una aseveración importantísima: ningún ordenador puede reemplazar al maestro de carne y hueso. El contacto humano constituye el elemento más importante en el proceso de aprendizaje y, si bien el ordenador es el recurso educativo más eficaz que se conoce, no puede sustituir la enseñanza afectuosa impartida por el maestro.

Los ordenadores proporcionan un gran entretenimiento, por lo que resulta una buena idea que se permita a los niños practicar juegos con ellos. A muchos padres les preocupa la posible alienación que pueden producir en el niño juegos como los "invasores del espacio" o Pac-Man, pero aunque dichos juegos son muy divertidos, no existe la más mínima evidencia que sugiera que el atractivo que ejercen sobre los niños rebase los límites de la simple fascinación.

La lógica del LOGO

Mostramos aquí cómo se construyen formas en la pantalla mediante la utilización del lenguaje LOGO.
El LOGO es un lenguaje de ordenador desarrollado específicamente para que los niños pequeños (de 4 o 5 años) programen un ordenador. Este lenguaje fue elaborado a finales de la década de los sesenta en el Massachusetts Institute of Technology (Estados Unidos) por un equipo dirigido por Seymour Papert, un matemático que había trabajado con Jean Piaget, el famoso pedagogo, en el centro que éste regentaba en Ginebra.
Para los niños más pequeños, el LOGO se convierte en una "tortuga", ya sea en forma de robot mecánico o como un triángulo luminoso en la pantalla de un ordenador. La orden FORWARD 10 hace que la tortuga se desplace 10 unidades hacia adelante, dibujando una línea. La orden RIGHT 90 determina que la tortuga trace un ángulo recto. Mediante secuencias de órdenes puede conseguirse que la tortuga dibuje cuadrados, triángulos, círculos e incluso formas menos convencionales. Asimismo, es posible enseñar a la tortuga a "recordar" las órdenes. Los niños, sin advertirlo, mientras enseñan a la tortuga no hacen más que programar un ordenador

Modo de dibujar un cuadrado:

```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
```

Cómo construir una orden «BOX» (recuadro):

```
TO BOX
REPEAT 4 (FORWARD 50 RIGHT 90)
END
```

La orden STAR (estrella):

```
TO STAR
REPEAT 12 (BOX RIGHT 30)
END
```

Pueden construirse otras formas muy fácilmente mediante la combinación de la orden BOX con otras similares



Fatiga visual

Algunas personas que trabajan frente a pantallas de ordenador sufren de fatiga visual; pero el período en que los niños permanecen frente a la pantalla no es tan prolongado como para que surja este problema. En el caso de los adultos, el problema parece ser de "acomodación", un fenómeno que se produce cuando el ojo permanece fijo a una distancia focal determinada y precisa de un cierto tiempo para reacomodarse. Si usted advierte que su hijo se concentra demasiado en el ordenador, mirando fijamente a la pantalla durante largos períodos, el problema podrá solucionarse haciéndole que tome un breve descanso cada quince minutos.
Es necesario mencionar un problema que plantean los televisores antiguos. Se ha descubierto que algunos televisores en color fabricados antes de 1970 emiten una dosis de baja radiación que puede resultar peligrosa si se trabaja regularmente muy cerca de ellos. Si decide utilizar su antiguo televisor en color para trabajar con el ordenador, es conveniente que compruebe si ha sido fabricado después de 1970

Los niños menores de siete años necesitan ayuda y supervisión para encender el ordenador y el televisor, y cargar un programa. Si éste es bastante bueno, se puede dejar que lo utilicen por sí solos, aunque esto dependerá de su destreza en la lectura y de su capacidad para responder a las preguntas del programa. Las exigencias en cuanto al hardware son muy sencillas. Un microordenador debe ser sólido: los niños golpean el teclado con los puños cerrados, tiran de las conexiones eléctricas y constantemente tocan la pantalla. Si el sistema es frágil, las conexiones inseguras o resulta difícil de utilizar, no despertará el interés de un niño pequeño. Algunos expertos consideran que el teclado idóneo de un ordenador para niños de corta edad debe ser grande y con teclas claramente definidas. Sin embargo, cuando el niño ya ha desarrollado por completo su capacidad motriz (generalmente a los siete años), es capaz de manipular teclados que incluso resultarían difíciles para un adulto. Los teclados sensibles al tacto de los microordenadores Sinclair más baratos no resultan convenientes para niños menores de 9 o 10 años; por el contrario, las versiones mayores del teclado impreso, como el del Philips Videopac 7000, son adecuadas incluso para niños de cuatro años de edad.

Jóvenes programadores

Para los niños de corta edad la elección del software es más complicada. Si proyecta utilizar un sistema basado en cassettes, usted deberá supervisar el proceso de carga y almacenamiento por completo. Por el contrario, si el sistema que utiliza es a base de discos, comprobará que sus hijos pueden manejar de forma correcta los discos flexibles. Para los más pequeños, niños menores de siete años, uno de los mejores dispositivos para almacenar programas es el cartucho ROM, una caja de plástico que contiene un circuito integrado con un programa fijado eléctricamente. El inconveniente de los sistemas de este tipo consiste en que no permiten al usuario almacenar ningún trabajo; sin embargo, los cartuchos son virtualmente indestructibles, por lo que los niños pueden utilizar el ordenador sin ninguna supervisión.

Si decide adquirir un ordenador exclusivamente para sus hijos, intente colocarlo en un lugar determinado, ya que trasladarlo de una habitación a otra, con la conexión y desconexión de cables que ello supone, si bien no dañará el ordenador (a menos que se le caiga al suelo), posiblemente haga que el niño renuncie a todos esos preparativos y opte por una actividad menos conflictiva, como encender el televisor, por ejemplo.

El centro de trabajo ideal

En una situación ideal, el ordenador del niño debería instalarse en su habitación, junto a su propia pantalla de televisión. Si usted está decidido a que la actitud de sus hijos hacia los ordenadores sea positiva, debe considerar la posibilidad de instalarles un centro de trabajo en alguna de sus habitaciones y adquirir un televisor de segunda mano para su uso exclusivo. (Instale la central de ordenadores en la habitación de su hijo mayor. Es posible que él o ella decidan utilizar el ordenador cuando sus hermanos se hayan ido a dormir.) Un televisor en blanco y negro antiguo puede comprarse por muy poco dinero y, siempre que disponga de sintonizador de canales, resultará adecuado para visualizar en pantalla la información de un ordenador.

Existe una amplia gama de argumentos acerca del efecto del color en los ordenadores destinados a niños pequeños; para algunos especialistas el color constituye un elemento vital, mientras que para otros es sólo un atractivo adicional, aunque innecesario. Resulta obvio que si debe optarse entre una conexión permanente a un televisor en blanco y negro en el dormitorio de los niños, y una conexión temporal al televisor en color de la sala de estar familiar, la instalación de carácter permanente es, indudablemente, la más acertada.

Si usted puede instalar un centro de trabajo/juego con ordenador, ya sea permanente o semipermanente, en una de las habitaciones de sus hijos, constituye una idea acertada disponerlo todo de modo que sea posible trasladar el ordenador sin alterar la instalación.

Cuando proceda a instalar la mesa para el ordenador, asegure los cables y conexiones, de modo que los niños no los arranquen accidentalmente. (Asegúrese asimismo de que todos los enchufes estén bien protegidos y cubiertos por cinta aislante para que no ofrezcan peligro. Y cuide que el televisor no esté conectado a la antena; de lo contrario, puede aparecer "accidentalmente" en la pantalla la última película, cuando los niños se hayan ido a la cama.) Es muy importante que el ordenador repose en una base estable y no oscile hacia ningún lado. La Sinclair ofrece una bandeja que mantiene en posición fija su ordenador Spectrum, que es muy ligero; si sus hijos son muy traviesos, idee algún sistema, una abrazadera o algo similar, para fijar el microordenador. Por supuesto, el ordenador que usted adquiera para sus hijos también será utilizado por toda la familia, de modo que, en ese caso, es mejor comprar un duplicado de los cables y (de ser necesario) un segundo paquete de alimentación principal. Éstos son relativamente baratos y le permitirán darles las buenas noches con firmeza a sus hijos, desenchufar el ordenador y la grabadora (o la unidad de disco) de su centro de trabajo en el dormitorio (dejando todos los cables enrollados en el mismo lugar), y enchufarlos en su propio televisor utilizando los duplicados de los cables.

El Big Trak

Aunque parece un carro de combate de juguete, en realidad se trata de un eficaz instrumento de aprendizaje. El Big Trak es un juguete programable mediante un ordenador, que permite al niño planificar con toda precisión los movimientos que desea que efectúe el carro. Este pequeño vehículo es capaz de memorizar hasta 16 pasos y se puede programar para que deambule de una habitación a otra de la casa antes de regresar a su base. Mientras el niño se divierte, el ordenador le ayuda a explorar su mundo físico y a elaborar los pasos individuales necesarios en un programa sencillo para ordenador. Pese a su aspecto agresivo, el Big Trak es del agrado tanto de niños como de niñas



Rizando el rizo

En esta segunda parte del curso nos ocuparemos de cómo interrumpir un bucle, cómo volver a él una determinada cantidad de veces y de la numeración de las líneas

La primera parte de nuestro curso de programación BASIC terminaba con el programa listado abajo. El programa funcionaba bien pero, en virtud del GOTO de la línea 70, efectuaba una y otra vez un bucle hasta el principio y no terminaba nunca. La única manera de salir del bucle era utilizando la tecla BREAK o la tecla RESET.

Ahora vamos a conocer uno de los modos de salir de un bucle como éste, incorporando una "comparación" en el programa. ("Comparación" es la acción de fijar una condición, para luego comprobar si existe.) La forma habitual de hacerlo consiste en probar con un número que en realidad jamás desearíamos utilizar en el programa. El programa nos permitía digitar un número que luego el ordenador imprimía en la pantalla después de sumarle 1. Podemos decidir que nunca desearíamos dar entrada a un número mayor que 999. En este caso hemos de comparar para ver si el número al cual le hemos dado entrada es mayor que 999. Digite el programa y luego agregue:

```
35 IF A > 999 THEN GOTO 80<CR>
```

Ahora ponga en marcha otra vez el programa y verá que funciona como antes; a menos, claro está, que dé entrada a un número mayor que 999. Pruebe digitando 1000<CR> y vea qué ocurre.

¿Por qué esta vez el programa se detuvo? La causa está en el IF (si) de la línea 35. Cuando el BASIC encuentra una expresión IF, sabe que a continuación viene una comparación lógica. El signo > significa "mayor que". Por lo tanto, la línea 35 significa IF (variable) A (es mayor que) 999 THEN GOTO (línea) 80 ("si la variable A es mayor que 999, luego volver hasta la línea 80"). Si usted digita 1 000, el valor de A se convierte en 1 000, que es mayor que 999, de modo que el programa "luego vuelve hasta" (THEN GOTO) la línea 80, que es el final del programa. Si A no fuera mayor que 999, la parte THEN (luego) de la línea sería ignorada y el programa continuaría con la línea siguiente.

De modo que al hacer funcionar este programa, usted puede dar entrada a todos los números que desee, con la condición de que no sean mayores que 999. Tan pronto como usted dé entrada a un número mayor que 999, la sentencia IF-THEN lo detecta y concluye el programa yendo hasta END (final). Cuando un programa BASIC llega hasta el final o se termina, se le dará a usted un aviso de "listo" en pantalla. Según sea su ordenador, este aviso tendrá diversas formas. En el Dragon, el aviso es OK. En el Sord, READY. Cualquiera que sea la forma asumida, el aviso de listo es la manera que tiene el BASIC de decirle a usted que no hay ningún programa funcionando y que está a la espera de sus órdenes.

Existen muchas variantes en la forma en que las distintas versiones de BASIC utilizan THEN. Consulte el recuadro de "Complementos al BASIC" en la página 39.

Otras comparaciones que se utilizan en BASIC son < (menor que), = (igual a), >= (mayor o igual a), <= (menor o igual a) y <> (distinto a). A medida que el curso avance, iremos encontrándonos con frecuencia con estas comparaciones.

Antes de seguir adelante, es conveniente realizar algunos ejercicios para acostumbrarse a utilizar estas comparaciones.

Ejercicios

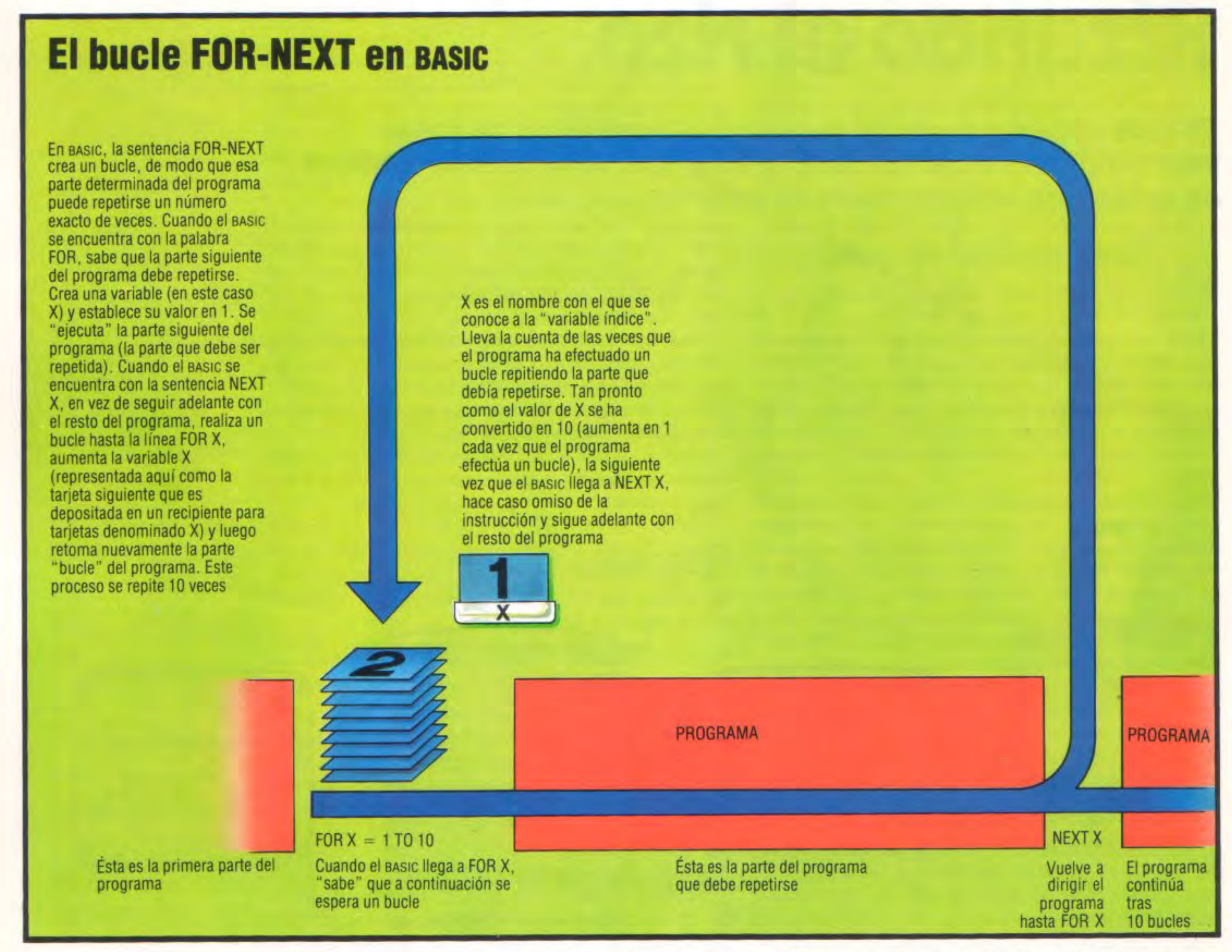
- Cambie una de las líneas de modo que el programa se interrumpa si A = 1000.
- Cambie una de las líneas de modo que el programa se interrumpa si el número al cual le da entrada es menor que cero.
- Cambie la línea GOTO de modo que haga que el programa efectúe un bucle hasta el principio si A es igual a o menor que 500. Una pista: no necesitará una línea IF-THEN y una línea GOTO separadas.

Descubriendo FOR-NEXT

En muchas ocasiones al escribir un programa usted deseará repetir algunos puntos del mismo un determinado número de veces. El GOTO de la línea 70 hacía que el programa efectuara un bucle todas las veces que lo deseáramos. Luego agregamos una sentencia IF-THEN en la línea 35 que nos permitía escaparnos del bucle dando entrada a un número "fuera de escala".

```
10 REM LOS ORDENADORES NUNCA SE EQUIVOCAN
20 PRINT "DIGITE UN NUMERO"
30 INPUT A
40 LET A = A + 1
50 PRINT "CREO QUE EL NUMERO DIGITADO ERA ";
60 PRINT A
70 GOTO 20
80 END
```

El bucle FOR-NEXT en BASIC



Sin embargo, tal como aprendimos en la primera parte del curso, en algunas ocasiones utilizar GOTO para hacer un bucle no es adoptar el procedimiento más acertado.

Volvamos a nuestro antiguo programa, ahora modificado para que se ajuste a la verdad, para multiplicar por 10 la entrada del número, y hagámoslo exactamente ocho veces.

```
10 REM MULTIPLICADO POR 10
20 FOR X = 1 TO 8
30 PRINT "DIGITE UN NUMERO"
40 INPUT A
50 LET A = A* 10
60 PRINT "SU NUMERO MULTIPLICADO POR 10 ES";
70 PRINT A
80 NEXT X
90 END
```

Digite este programa, lístelo (LIST) para verificar que no haya ningún error y luego hágalo funcionar (RUN). Se le solicitará un número sólo ocho veces. Luego el programa se detendrá. La causa de esta detención debemos hallarla en la línea 20.

```
20 FOR X = 1 TO 8
```

Esta línea forma parte de un bucle FOR-NEXT. Este bucle constituye una de las más útiles estructuras que ofrece el lenguaje BASIC. Merece que la estudiemos atentamente.

Tal como la hemos utilizado aquí, hemos creado una variable denominada X. (En la primera parte del curso, p. 21, hemos explicado qué es una variable.) Podríamos haberla llamado de cualquier manera (excepto A, que ya la estamos utilizando con otro fin). FOR siempre debe utilizarse seguido de NEXT, pero NEXT aparecerá más tarde en el programa (después del trozo que ha de repetirse). El elemento FOR de un bucle FOR-NEXT tiene siempre la siguiente forma:

FOR variable = valor inicial TO valor final
(*para* variable = valor inicial *hasta* valor final)

En nuestro ejemplo FOR X = 1 TO 8 hemos llamado X a la variable y le hemos dado un valor inicial de 1. El ordenador ejecuta entonces la siguiente parte del programa; el número que digitamos es multiplicado por 10 y luego impreso en la pantalla. Después de esto llegamos a NEXT X y el programa hace un bucle volviendo hasta donde está la variable X, es decir, a la línea 20. Después de hacerlo aumenta X en 1, de modo que X adquiere un valor de 2. Luego se vuelve a ejecutar la parte del programa comprendida entre el bucle FOR-NEXT. Al llegar nuevamente hasta NEXT, en la línea 80, el programa vuelve a hacer un bucle y convierte a X en 3.

El programa continúa repitiéndose de esta manera hasta que X se convierte en 8. Entonces el bucle se termina; NEXT X ya no vuelve hasta FOR X y el programa continúa en la línea siguiente.

Otros usos de los bucles FOR-NEXT

Con frecuencia los bucles FOR-NEXT se utilizan para producir dilaciones en el programa. En algunas ocasiones a usted no le interesará que todo se realice a la máxima velocidad; en estos casos, puede introducir una "demora". Probablemente habrá advertido que en el programa MULTIPLICADO POR 10 las respuestas eran transmitidas de modo tan rápido que parecían instantáneas. Hagamos que el ordenador parezca tomarse un tiempo para pensar antes de dar las respuestas; para lograrlo, insertaremos una demora valiéndonos de FOR-NEXT. Agregue en su programa las líneas que aparecen en color azul.

```
10 REM MULTIPLICADO POR 10
20 FOR X = 1 TO 8
30 PRINT "DIGITE UN NUMERO"
40 INPUT A
50 LET A = A* 10
52 FOR D = 1 TO 1000
54 NEXT D
60 PRINT "SU NUMERO MULTIPLICADO POR 10 ES";
70 PRINT A
80 NEXT X
90 END
```

Hemos agregado otras dos líneas, 52 y 54, en nuestro bucle FOR-NEXT original. Ahora examinémoslas.

```
52 FOR D = 1 TO 1000
54 NEXT D
```

D se establece en 1 y el programa continúa con la línea siguiente. Ésta es la sentencia NEXT correspondiente. En realidad, en el interior del bucle no sucede nada; el programa se limita a efectuar un bucle hasta la línea 52 y aumenta D a 2. Esto ocurre 1 000 veces antes de que el programa continúe con la parte siguiente, que está imprimiendo la respuesta. Los ordenadores son veloces, pero para todo se requiere un tiempo finito, de manera que efectuar 1 000 veces un bucle lleva una cantidad de tiempo considerable. Los ordenadores difieren en cuanto al tiempo que invierten en efectuar un bucle. En el Epson HX-20, este bucle FOR-NEXT tarda 2,9 segundos, mientras que en el Spectrum tarda 4,5 segundos. Experimente cambiando el número que utiliza como límite máximo en la línea 52.

Para hacer que el ordenador se comporte más como un ser humano, agregue estas tres líneas:

```
56 PRINT "AHORA DEJEME VER..."
57 FOR E = 1 TO 1000
58 NEXT E
```

Liste (LIST) el programa y hágalo funcionar (RUN). Ahora tenemos dos demoras cuya función es sólo dilatar el tiempo del programa.

Agregue estas dos líneas:

```
51 REM ESTE BUCLE PIERDE EL TIEMPO
55 REM ESTE PIERDE AUN MAS TIEMPO
```

Ahora liste (LIST) el programa y examínelo con atención. Observe cómo todas las líneas extras que hemos agregado se han situado exactamente en los lugares precisos. Lo que nos lleva al último punto de esta parte del curso: los números de las líneas.

Comenzamos nuestro programa original con la línea 10 y continuamos saltando de 10 en 10 para cada nueva línea, terminando con la línea 90. Podríamos haber elegido otros números cualesquiera, por ejemplo 1, 2, 3... 9. Pero, de haberlo hecho, ¿cómo habríamos podido intercalar las líneas extras? A los programadores siempre se les ocurren ideas nuevas y mejores que introducir, de modo que dejemos lugar para ellas en estos grandes huecos entre los números de las líneas en las versiones "Mark I" de sus programas. Si lo desea, puede comenzar incluso por la línea 100 e ir saltando de 50 en 50 o de 100 en 100.

Algunas versiones de BASIC incluyen una orden muy útil denominada AUTO. El BASIC del Epson HX-20 la tiene. No obstante, el Dragon 32, los ordenadores Sinclair y el Commodore Vic 20 no cuentan con ella. Si su BASIC posee AUTO, puede ahorrarse mucho tiempo al hacer que los números de las líneas se generen automáticamente. Averigüe si su BASIC posee AUTO digitando:

```
AUTO 100, 10<CR>
```

Si su BASIC posee AUTO, en su pantalla se verá:

```
100
```

La pantalla muestra el número 100 seguido de un espacio y luego el *cursor*. El cursor es una marca (algunas veces una línea, otras un cuadrado) que muestra en la pantalla dónde aparecerá el carácter siguiente. Usted puede comenzar a entrar la primera línea del programa desde la posición del cursor. Cuando usted pulse <CR> la línea siguiente aparecerá automáticamente, comenzando por el número de línea 110. El AUTO, si su ordenador dispone de él, puede utilizarse tanto solo como con uno o dos "argumentos". La palabra *argumento* es un término matemático. En la expresión $2 + 3 = 5$, los argumentos son 2 y 3. La orden AUTO puede utilizarse sola (por ejemplo, AUTO<CR>), o bien con un argumento (por ejemplo, AUTO 100<CR>) o con dos argumentos (por ejemplo, AUTO 300,50). AUTO empleado solo suele hacer que los números de línea comiencen a partir de 10 y sigan saltando de 10 en 10. Si se utiliza sólo un argumento (por ejemplo, AUTO 100<CR>), el primer número será 100 (en este caso) y luego los números irán saltando de 10 en 10. Si se utilizan dos argumentos, la primera cifra especifica el número de línea inicial y la segunda el número según el cual se irán produciendo los aumentos. AUTO 250,50<CR> dará 250 como número inicial, el número siguiente será 300 y así se irá aumentando, de 50 en 50. Incluso en el más sencillo de los micros, es prácticamente imposible que se dé el caso de que usted se quede sin líneas.

En el próximo apartado de este curso nos ocuparemos especialmente de las diversas maneras en que se puede mejorar la presentación visual de un programa en la pantalla y de los distintos modos de imprimir información.



El código de barras

Esas misteriosas barras sobre las tapas de los libros, en las cubiertas de revistas o fascículos y en los productos de los supermercados transmiten su mensaje a un ordenador y ayudan a llevar el control de almacenaje y distribución más eficazmente

Decodificación de barras

"7" "2"

COMIENZO FIN

La ilustración superior muestra un código de barras, que representa el número 72. A simple vista parece sólo una serie de líneas negras de distinto grosor. En este caso cada serie incluye cinco barras, dos de las cuales son anchas. La posición de las dos líneas anchas en cada serie indica el número. Unas barras adicionales señalan el principio y el final de cada unidad de información individual. En esta ilustración, el número 72 está codificado en dos unidades, el dígito 7 y el dígito 2.
Existen muchas formas diferentes de codificar información en un código de barras. Dado que el grosor de la barra es variable, ésta puede representarse numéricamente como un 1 o un 0. Esto nos conduce directamente a la matemática binaria de los ordenadores. El *Universal Product Code* es un código de barras ligeramente diferente que se utiliza en el comercio. Según este código las barras pueden ser de diversos espesores, se requieren menos barras y la información se lee a partir de la anchura de la línea.
La razón de que los códigos de barras se haya extendido tanto en las librerías y en los supermercados se debe a que pueden ser leídos por una máquina. El librero se vale de un "lápiz" sensible a la luz, que ilumina el código de barras y registra la cantidad de luz reflejada. En contraste con el fondo blanco, las barras negras casi no reflejan luz. La luz reflejada se convierte en una señal eléctrica que es amplificada. Cuando se registra luz se traduce en un 1 binario, y si no se registra luz y por lo tanto no se produce señal, en un 0 binario.
Las barras proporcionan secuencias de ceros, y cuanto más anchas son, más ceros contienen. Del mismo modo, el fondo blanco proporciona secuencias de unos. De esta manera, la "varita mágica" alimenta el ordenador con los patrones de dígitos binarios, a partir de los cuales puede determinar la composición de un código de barras

¿Se ha preguntado en alguna ocasión qué significan esas barras impresas en los productos empaquetados del supermercado, en las tapas de un libro o en las cubiertas de algunas revistas o fascículos? Pues bien, se trata del *código de barras*, un ingenioso recurso que puede ser leído en una fracción de segundo por una "varita mágica" sensible a la luz y que introduce en un ordenador la información relativa a esos productos. Mediante este sistema se puede saber, en cualquier momento, cuál es el stock, la fecha de fabricación o impresión, la fecha de caducidad, etc., de un determinado producto.

Veamos cómo funciona este control en el caso de los libros de bolsillo. Cada libro que se publica en los principales países del mundo posee un International Standard Book Number (ISBN), que consiste en uno o más dígitos en los cuales constan el idioma o la zona geográfica en los que se publica el libro (84 para España y todos los países de lengua castellana; el 0 y el 1 para Gran Bretaña, Estados Unidos y otros países de lengua inglesa; el 2 para Francia; el 3 para Alemania, Austria y la Suiza de lengua alemana, etc.), entre dos y siete dígitos para identificar al editor, y entre uno y seis dígitos para identificar el título del libro y su edición. Esto arroja un total de nueve dígitos, y luego existe un dígito de control (que el ordenador utiliza para verificar si todos estos dígitos le han sido proporcionados en el orden correcto).

Para la codificación en barras, los libros se numeran de acuerdo al sistema European Article Numbering (EAN; numeración europea de artículos), que utiliza un total de 13 dígitos (la mayoría de los productos de las tiendas de comestibles suelen utilizar un número corto de ocho dígitos). Los tres primeros dígitos corresponden a la "bandera" de la EAN (para los libros es el 978). Luego viene el ISBN y, por último, un dígito de control EAN alternativo.

En España, el ISBN (junto con su propio dígito de control) también se imprime en números en el código de barras; éstos pueden ser leídos tanto por el ojo humano como por un lector de caracteres ópticos (Optical Character Reader).

Los lectores de caracteres ópticos constituyen otro desarrollo interesante de consecuencias mucho más amplias. En la actualidad, existen máquinas que son capaces de leer literalmente la palabra impresa mediante la exploración óptica de la línea. La señal de salida del lector se acopla al ordenador, el cual puede entonces procesar la información de diversas maneras. Las palabras leídas por el explorador pueden, por ejemplo, visualizarse en la pantalla del ordenador, eliminando la agotadora tarea de mecanografiarlas. Y todo ello gracias a este ingenio que está revolucionando el mundo: el ordenador.



Ian McKinnell

La revolución del ordenador



Ordenadores en la granja
Una empresa dedicada a la realización de software ha diseñado un programa llamado "Optimizador", que minimiza el coste del pienso para una granja porcina. La proporción de cereales varía a diario en función del tipo de alimentación. Usando el ordenador, el granjero da las cantidades necesarias diarias a los cerdos y el valor del nutriente de los cereales disponibles en términos de proteínas, energía y vitaminas. El programa determina entonces la proporción de cereales más económica. Este software ayuda al granjero a calcular las mezclas de piensos más eficientemente. Persisten, no obstante, unos ciertos problemas de "gusto". Los animales prefieren una dieta regular, y son reacios a tomar una comida completamente distinta a la anterior. El programa, a pesar de todo, ha obtenido un gran éxito mundial y ha sido vendido desde Thailandia a México

El ordenador puede ahorrar tiempo y esfuerzo a las pequeñas empresas y multiplicar, al mismo tiempo, sus beneficios

El ordenador nació en los laboratorios de las universidades y centros militares. Las primeras máquinas se construyeron para el cálculo de las trayectorias de los proyectiles disparados, por ejemplo, desde un acorazado en medio de una tempestad, y para la predicción meteorológica.

No obstante, no tardaría mucho tiempo en ser evidente la utilidad de los ordenadores en aplicaciones comerciales. Inicialmente, sólo las grandes compañías podían disponer del capital necesario para su mecanización. Pero la revolución de la microelectrónica a finales de los años setenta puso el ordenador al alcance de la pequeña empresa.

¿Cómo puede un ingenio creado para fines militares ayudar, por ejemplo, a una papelería, dedicada a la venta de objetos de escritorio, periódicos y revistas? El ordenador puede admitir y almacenar gran cantidad de datos. Asimismo, puede reorganizar la información que le ha sido proporcionada de forma más útil. En una papelería-librería suele haber grandes existencias de diversos artículos, y su control resulta una tarea harto pesada.



Las cifras de ventas son introducidas en el ordenador y un programa controla el stock de cada artículo. En el momento en que las existencias de cualquier artículo son menores que el stock mínimo, el ordenador genera un mensaje para que se realice el pedido correspondiente. La memoria del ordenador puede efectuar también un pedido estándar. Entonces, el vendedor añade los detalles particulares de ese artículo, y el ordenador imprime la hoja de pedidos.

Pero el ordenador no se detiene ahí. Las largas y tediosas tareas de cálculo de nóminas, impuestos, confección de cuentas de fin de ejercicio, etc., caen dentro del ámbito de su aplicación.

Los ordenadores no son sólo prácticos, sino a veces totalmente indispensables. Sin su ayuda, el hombre nunca hubiera puesto el pie en la Luna. Existen problemas en el diseño de cohetes y en la navegación que sólo pueden ser resueltos mediante ordenadores. Gracias a ellos, la humanidad tiene acceso a campos de la ciencia y la técnica que antes le estaban vedados. Pero para el propietario de un pequeño negocio, que se pregunta qué puede hacer un ordenador que no pueda efectuar él personalmente, la respuesta es sencilla: dinero. Actividades que hasta ahora precisaban más dinero en organización y administración del que producían, son rentables con la llegada del microordenador.

Supongamos que al principio la orientación de este negocio era la venta de periódicos y revistas. Pero el escaso margen de beneficio obtenido llevó a su propietario a comerciar en otros artículos. Ahora el ordenador puede reportarle beneficios al mantener un control de las demandas diarias de cada cliente. Los clientes pueden cambiar de periódico cada día sin trastocar el sistema de pedidos. A la distribuidora se le solicita el número exacto de periódicos o revistas necesarios. A fin de mes, el ordenador calcula la cuenta de cada cliente e imprime la factura adecuada. Gracias al ordenador, muchos negocios vuelven a ser rentables.

El sistema para la pequeña empresa



Un ordenador apropiado para una pequeña empresa debe ser una máquina de gran calidad, capaz de soportar muchas horas de uso diario. Es esencial un buen teclado y una unidad de disco para realizar los programas de oficina. El ordenador NCR ilustrado aquí es un sistema típico. El único gasto accesorio requerido es el coste de la impresora

Contabilidad

Todas las cuentas de la empresa pueden realizarse con un software especial de contabilidad. El diario de ventas, los libros de caja o de doble entrada pueden automatizarse para producir las cuentas mensuales, trimestrales o anuales en un formato adecuado para contables o auditores.

El mejor software de contabilidad comprende "módulos" que trabajan a la vez, de forma que los nuevos datos suministrados por una sección de este software (las salidas de caja, por ejemplo) actualizan el resto de las secciones del sistema de contabilidad. La ventaja para el empresario se traduce en pocas horas-hombre dedicadas a asentamientos rutinarios en los libros, cuentas más exactas y altos beneficios al final de la jornada

Nómina

Un programa de nómina calcula el salario e imprime las hojas de nómina de todos los empleados de la empresa. Estos programas deben tener en cuenta las horas trabajadas, el salario base, las deducciones típicas, incluyendo impuestos y seguridad social, etc.

Dichos programas funcionan normalmente haciendo el operador una serie de preguntas sencillas: ¿Cuántos empleados hay en la nómina?, ¿Cómo se pagan los salarios, semanal o mensualmente?, etc. Estas preguntas se contestan digitando la información en el teclado del ordenador

Control de stocks

Un programa de control de stocks proporciona al operador del sistema información sobre su variación. El programa puede, incluso, efectuar automáticamente el pedido de los artículos cuyo nivel de existencias esté por debajo del mínimo. Algunos programas pueden obtener los datos de entrada mediante un lector de barras óptico (véase p. 40), en el momento en que es vendido el artículo, de forma que el precio puede ser visualizado en la caja registradora y al mismo tiempo actualizarse el inventario

Facturación

Un programa de facturación imprime las facturas con menos problemas y más exactitud que si se realiza de forma mecánica. Al hacer funcionar un programa de este tipo, se le pregunta al operador una serie de cuestiones, y éste, mediante el teclado, introduce las respuestas apropiadas. El programa consulta los registros para comprobar si la factura es legítima y los detalles son adecuados. El programa también intercambia datos con el control de stocks para mantener actualizadas las existencias


Tony Sleep

Bajo el capó
Casi todos los negocios pueden beneficiarse de un ordenador. Cuando un coche entra en el taller para un servicio de mantenimiento, el ordenador suministra una relación de las verificaciones que deben hacerse. El mecánico trabaja según estas instrucciones, introduciendo en el ordenador la clave de cada recambio y el tiempo empleado en cada reparación. Al final, el ordenador facturará el valor de las piezas y mano de obra. Los ordenadores se ocupan de las labores rutinarias para dejarle tiempo libre para otros trabajos más creativos

BITS Y GASTRONOMÍA

Parecía que el restaurante que tenía Jordi en el elegante barrio de Pedralbes, en Barcelona, tenía bastante éxito; normalmente se encontraba muy concurrido. No obstante, el dueño se hallaba disconforme con los gastos generales y sobresueldos, que había que restar de la recaudación.

Un ordenador le proporcionó algunas respuestas. Recogiendo y despachando los pedidos y expidiendo detalladas cuentas para los clientes, la máquina minimizó los errores.

El propietario quiso conservar las gratificaciones al personal, y esto requirió un análisis de productividad. Por este motivo, decidió adquirir un ordenador y software que incluyera un programa de nómina y otro de control de existencias y de coste de las diversas recetas.

EN LA PELUQUERÍA

En una elegante peluquería de Barcelona, un microordenador formula a la cliente algunas preguntas de tipo personal. La mujer es interrogada acerca de sus gustos o preferencias sobre el acondicionador, tratamiento de color, permanente, etc.

De este modo, el peluquero puede trabajar a partir de las precisas instrucciones del ordenador, que especifica un producto determinado y cómo debe ser usado. La cliente recibe un programa personal para el tratamiento de su cabello, de acuerdo con las respuestas proporcionadas al ordenador.

Los programas de diagnóstico son simples de operar y tardan pocos minutos en aparecer. El dueño de la peluquería opina que, gracias a la ayuda del ordenador, puede realizar teñidos de cabello de alta calidad, y del gusto de su numerosa clientela.

PAN Y ORDENADOR

Un día una familia decidió instalar una panadería en el barrio de Salamanca, en Madrid. Al poco tiempo de abrir el negocio, los propietarios se dieron cuenta de la disparidad de los pedidos: 20 bollos por aquí, 400 barras por allá, y de que esta contabilidad podría llevarla muy bien un ordenador.

Ahora, la flamante máquina puede incluso facturar semanalmente, hacer la nómina y calcular las remesas, el coste de la harina y demás ingredientes, descuentos, etc.

Introducir 350 pedidos en la memoria del ordenador fue una labor ardua. Pero una vez esta tarea estuvo terminada, el microordenador pudo trabajar solo durante todos los fines de semana, facilitando sobremanera la administración del negocio.

Existen muchos otros paquetes de software para empresas. Los programas de tratamiento de textos están entre los más solicitados. Todas las correcciones realizadas mediante un programa de tratamiento de textos pueden efectuarse en una pantalla. Cuando el texto ha sido completado, se imprime una copia perfecta tantas veces como sea necesario. Aunque la repetición sea aburrida para una persona, no lo es para un ordenador.

El tratamiento de textos es un área muy apropiada para el ordenador. Con los avances tecnológicos, cada vez serán más las personas que dispondrán de un ordenador en la oficina o en su casa. La comunicación directa entre ordenadores reemplazará finalmente al envío por correo de facturas o pedidos. Los ordenadores serán en el futuro tan indispensables como lo es hoy el teléfono.

Pintando con números

Tres tipos fundamentales de gráficos de ordenador y cómo usarlos para crear imágenes

Un cuadro se compone de cientos de pinceladas aplicadas por el pintor sobre un lienzo desnudo. ¿Pero cómo puede un artista crear una imagen usando un ordenador?

Para las realizaciones gráficas, se emplean tres sistemas principales. El diseño gráfico debe tener en cuenta el grado de control sobre la resolución, o nitidez, de la imagen final. Cada microordenador utiliza gráficos de bloque, *pixels* o gráficos de alta resolución.

En los gráficos de alta resolución, el artista desea controlar los puntos luminosos del monitor. La única limitación es la capacidad de memoria del ordenador. La memoria modela la pantalla del televisor. Con un ordenador de 32 K de memoria, cada punto luminoso corresponde a una parte de la pantalla según el modelo del aparato.

En los gráficos de bloque, lo que el artista pierde en control sobre los puntos individuales que configuran la pantalla, lo gana en comodidad. Las formas elementales están ya programadas y se pueden obtener a través del software para construir una imagen. Se controlan directamente con el teclado y por lo general están situadas en la parte frontal de cada tecla. De esta forma, el teclado del ordenador se convierte en una paleta de pintor.

Cada forma está configurada dentro de una pequeña matriz de puntos: ocho filas por ocho columnas. Algunos microordenadores ofrecen además la posibilidad de definir los propios caracteres. Para definir el nuevo carácter y añadirlo a la gama del ordenador, se emplea un programa menor.

Los *pixels* (contracción de *picture cell*: unidad pictórica) se sitúan entre los gráficos de bloque y los de alta resolución. Éstos permiten al artista controlar cada pixel, que contiene más de un punto luminoso, pero

Las tres resoluciones



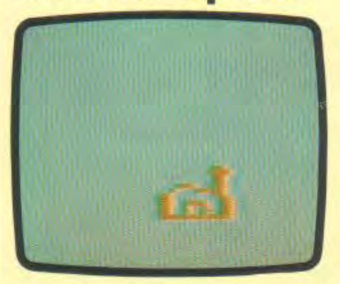
Los gráficos de bloque de baja resolución, de sólo 1 Kbyte de memoria, son suficientes para almacenar todos los detalles que deben aparecer en la pantalla. Si únicamente hay 40 bloques en cada fila y sólo 24 filas en la pantalla, el número total de bloques es 960, algo menos de un Kbyte (un Kbyte es igual a 1 024 bytes). Por tanto, un Kbyte de memoria para la pantalla puede almacenar ocho bits de información (255 combinaciones diferentes) para cada uno de los bloques de la pantalla.

En resolución media, el detalle que se obtendrá en la pantalla estará en función de la capacidad de la memoria disponible. Ocho Kbytes de RAM representan 65 536 puntos luminosos. Si el aparato es en color, una parte de la memoria se empleará para especificar el color de cada punto, reduciéndose la memoria para el número de puntos. Si se pueden reproducir ocho colores, se necesitarán tres bits para determinar el color de cada punto.

Un aparato de alta resolución requiere una gran cantidad de memoria. A veces, los aparatos de alta resolución tienen hasta 640 puntos por línea y la pantalla está formada por 240 líneas. Esto da un total de 268 800 puntos. Si sólo se va a representar un color, serán necesarios 33 600 bytes (268 800 dividido por 8). Como se ha dicho anteriormente, cada bit equivale a un cero o a un uno, y esto corresponde a que cada punto en la pantalla esté apagado o iluminado

Jo Lawrence/Ian McKinnell

Gráficos sprite






Sprites es el nombre que recibe la forma de lograr imágenes más naturales, y más fácilmente, en un ordenador. Originalmente desarrollados por Texas Instruments, los gráficos sprite pueden conseguirse en la actualidad en la mayoría de los ordenadores personales, incluidos el Commodore 64, los modelos Atari y el Sord M5. En los gráficos convencionales, las imágenes se componen en una pantalla simple, como si se dibujara en una hoja de papel. Con los sprites el ordenador tiene diversos planos o "capas", cada uno de los cuales contiene su propia imagen. En algunos ordenadores, como el Sord M5, pueden hacerse hasta 32 planos separados.
La manera más fácil de representar estos planos consiste en imaginarse láminas de plástico transparente. Si la lámina "más cercana" al observador contiene el dibujo de un árbol, mientras que la última incluye la imagen de una nube, ésta será vista pasar por detrás del árbol como si flotara en el cielo.
Al poner diversos elementos de la pintura en planos separados, podrán crearse efectos tridimensionales muy convincentes.
Los sistemas gráficos sprite tienen también muchas otras ventajas. La imagen "dibujada" en cada uno de los planos se llama, en el lenguaje de los ordenadores, *objeto*. Al crear un objeto (un pájaro, por ejemplo), el programador puede olvidar los detalles de cómo fue compuesto. Si quiere desplazarlo por la pantalla, puede especificar una velocidad y una dirección.

menos de ocho por ocho bloques. Cada pixel se puede hacer surgir individualmente y situar en la pantalla en la posición deseada.

En los gráficos de alta resolución se utilizan dos órdenes BASIC para realizar líneas: MOVE, para situar el inicio de una línea, y DRAW, para hacerla aparecer. Cada final de línea se identifica con un par de números, que representan la fila y la columna de la pantalla donde termina la línea. Normalmente, en la pantalla las filas se enumeran de arriba abajo, y las columnas, de izquierda a derecha. Por tanto, el punto situado en la fila cero, columna cero se encuentra en la esquina izquierda inferior de la pantalla.

El programa que presentamos a continuación puede hacerse en los ordenadores Lynx (aún no disponible en el mercado español) y BBC Micro. El ordenador debe conectarse en la función "gráficos". Digite el programa tal como aparece. Puede funcionar en cualquier ordenador de alta resolución con tan sólo reemplazar MOVE y DRAW por sus funciones equivalentes. (Consulte el recuadro "Complementos al BASIC" que aparece en esta misma página.)

```
10 MOVE 100,50
20 DRAW 92,95
30 DRAW 57,125
40 DRAW 100,110
50 DRAW 143,125
60 DRAW 108,95
70 DRAW 100,50
```

Cuando el programa funciona, en la pantalla debe aparecer un perfil "Trinacria". Este perfil recuerda la isla de Sicilia, cuyo antiguo nombre en latín es precisamente *Trinacria*. Este tipo de programa es apropiado para dibujar cualquier perfil formado por una retícula de líneas. Los números que se introducen con la orden DRAW indican la posición en la fila y la columna de cada punto que forma el perfil. Con estas funciones, se puede realizar cualquier tipo de dibujo. La única limitación es la resolución de la pantalla del ordenador y la perseverancia que tenga hasta el extremo de que incluso una curva puede representarse con puntos. Y las órdenes MOVE y DRAW proporcionan el acceso a tales puntos.

El siguiente programa realiza una imagen de un cono formado por una serie de círculos de sección variable. Tal como está escrito, este programa funcionará en el Sinclair Spectrum. Otros ordenadores, como el Dragon y el Oric, poseen además la función CIRCLE. (Consulte nuevamente el recuadro "Complementos al Basic".)

```
10 FOR K = 2 TO 40
20 CIRCLE 40 + K, 40 + K,K
30 NEXT K
```

Estos dos programas muestran cómo un ordenador puede realizar gráficos mediante funciones numéricas. No obstante, una digitación intermitente y desordenada conduce siempre al fracaso en el intento de llevar a cabo un dibujo continuo. Por ésta razón existe equipamiento especial para el diseño de imágenes. Éste puede conectarse al microordenador y así no es necesario introducir los miles de números que permiten obtener imágenes nítidas. El digitalizador es uno de estos aparatos. El artista efectúa su dibujo con un lápiz especial, y el digitalizador convierte el movimiento de éste en los números de filas y columnas que interpreta el ordenador.

Complementos al BASIC



Para introducir el programa en el BBC Micro, debe precederse de
5 MODE 0



Para introducir el programa en el Oric, debe precederse de
5 HIRES
y la línea 20 debe reemplazarse por
20 CURSET 40 + K, 40 + K, 0
25 CIRCLE K, 1

Para introducir el programa en el Dragon, debe precederse de
5 PMODE 4,1: SCREEN 1,1: COLOR 0,5:
PCLS
y la línea 20 debe reemplazarse por
20 CIRCLE (40 + K, 40 + K), K

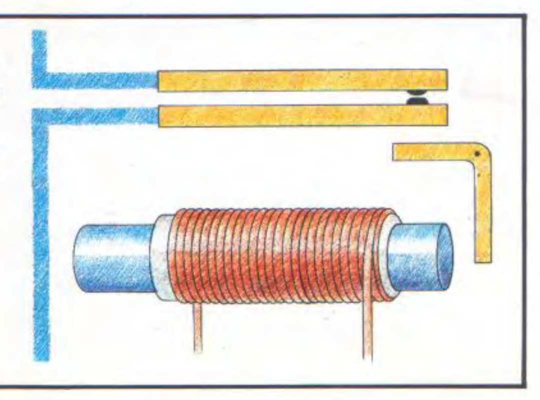
Conexión

Tras la revolución del ordenador se producen avances asombrosos en la «microingeniería»

El primer ordenador de válvulas
En 1943, en los momentos más críticos de la segunda guerra mundial, un coronel del Ejército de Estados Unidos solicitó una máquina calculadora para la artillería. El reto fue aceptado por la Universidad de Pennsylvania, que presentó su invento en 1946, habiéndose necesitado 7 237 horas-hombre para perfeccionarlo.
A la máquina se le dio el nombre de ENIAC (Electrical Numerical Integrator and Calculator) y fue el primer ordenador de válvulas. El ENIAC utilizaba 18 000 válvulas, 1 500 relés y emitía el calor equivalente a 200 kilovatios. Esta enorme construcción fue instalada en una sala de 9 × 30 m. Los primeros problemas fueron la escasa capacidad de memoria y la falta de fiabilidad. El ENIAC sólo podía almacenar 20 números de 10 dígitos y todo el programa tenía que hacerse reordenando las conexiones. En 1952 tuvieron que sustituirse más de 19 000 válvulas, debido a que la máquina sólo podía funcionar unos dos minutos antes de que las válvulas empezaran a fundirse. La vida del ENIAC fue corta, siendo retirado de funcionamiento en 1952

© Science Museum

Interruptor-relé
Al pasar la corriente a través del solenoide que rodea el núcleo de hierro, se crea una fuerza magnética. Esta fuerza atrae al ángulo de hierro, que gira sobre su eje situado en el vértice, une los dos contactos y cierra el circuito

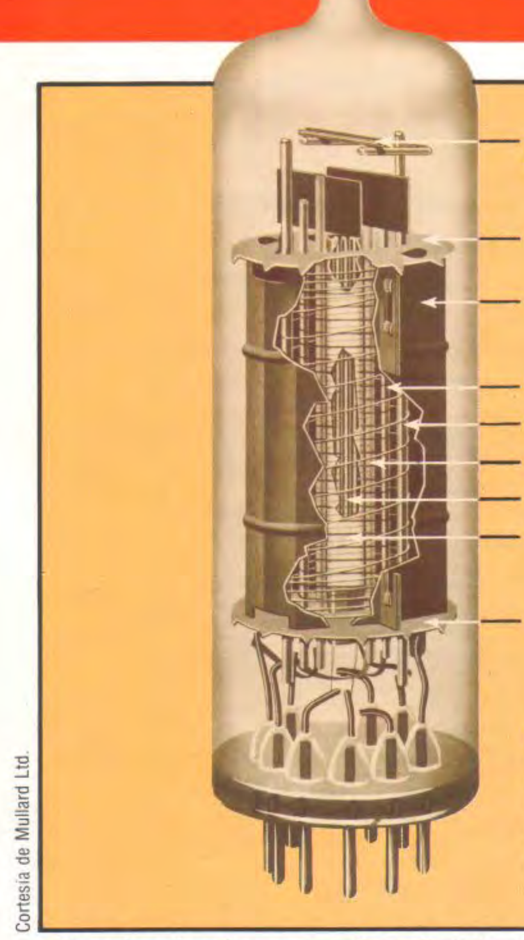


Tony Lodge

El ordenador moderno contiene millones de minúsculos interruptores electrónicos. Éstos son fundamentales en el diseño de los ordenadores; sin ellos no habría tenido lugar la revolución tecnológica que se ha producido a partir de la segunda guerra mundial.

En 1938, un ingeniero eléctrico, Claude Shannon, demostró que utilizando circuitos-interruptor eléctricos, podían realizarse operaciones lógicas. Desde que fue evidente que el funcionamiento de un ordenador consiste en una secuencia de operaciones lógicas, las investigaciones se dirigieron hacia la construcción de un interruptor electrónico.

El primer intento dio como resultado el relé. Este interruptor fue utilizado con éxito en los primeros pro-



Además de los tres componentes fundamentales (cátodo, ánodo y rejilla), muchas válvulas contienen un número adicional de elementos para mejorar su rendimiento. El principio de generación, sin embargo, permanece invariable

Cortesía de Mullard Ltd.

Válvula

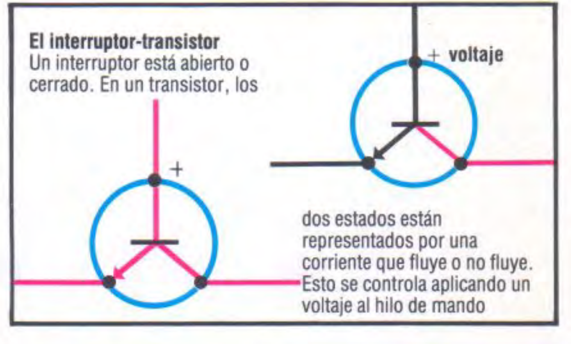
El tubo de gas de la ilustración contiene un terminal positivo y otro negativo (el ánodo y el cátodo), separados por una rejilla de alambre. Los electrones del cátodo son atraídos por el ánodo y producen una corriente a través de la válvula.
El flujo natural de electrones negativos hacia el terminal positivo es incrementado en dos formas. El cátodo es calentado hasta que se vuelve incandescente y su superficie es tratada con un compuesto químico especial.
La rejilla de mando está situada entre el cátodo y el ánodo y normalmente no interfiere el movimiento de los electrones. Pero si a la rejilla se le aplica una carga negativa, los electrones son repelidos y no alcanzan el ánodo. La fuerza de rechazo de la rejilla vence a la atracción del ánodo y se interrumpe el flujo de electrones. La corriente se para y, por tanto, el interruptor queda cerrado

totipos de ordenador, pero su diseño limitaba el desarrollo y aumento de potencia. El funcionamiento del relé no era totalmente eléctrico y sus componentes mecánicos fueron causa de frecuentes averías y de un funcionamiento lento, lo que creó desconfianza en cuanto a su fiabilidad.

La primera generación de ordenadores se caracterizó por el uso de válvulas termiónicas como componente esencial del interruptor. Estas válvulas eran de funcionamiento totalmente electrónico, y, por tanto, más rápido. Sin embargo, consumían grandes cantidades de electricidad (que era cara y causaba problemas con el calor generado), eran voluminosas y aún no del todo fiables.

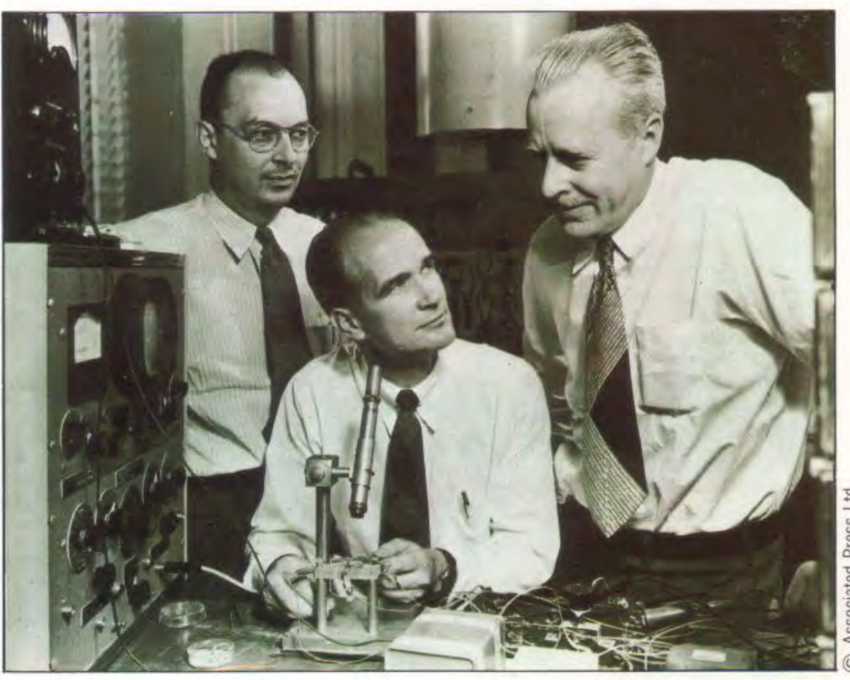
La invención del transistor anunció una nueva generación de ordenadores. Teóricamente, el funcionamiento del transistor es similar al de la válvula, pero su rendimiento es superior, es más pequeño y más barato de fabricar. Estas ventajas permitieron que el ordenador saliera de las universidades y los centros militares y entrara en el mundo comercial.

Los ordenadores actuales todavía utilizan transistores en la función de interruptores, pero éstos ya no son componentes independientes, separables. En un chip de silicio del tamaño de una uña, puede haber hasta un cuarto de millón de transistores, siendo cada uno de ellos tan pequeño que no puede distinguirse a simple vista. Aunque sean tan minúsculos, cada uno continúa siendo un interruptor. Al introducir los miles de interruptores necesarios para hacer funcionar un ordenador en el interior de un pequeño chip de silicio, se ha hecho posible lograr un ahorro drástico en los costes. Los ordenadores más caros y potentes de los años cincuenta, que ocupaban todo un laboratorio, han sido reducidos a un simple chip: la central eléctrica de los micros de hoy en día. Los ordenadores son ahora lo suficientemente pequeños y baratos como para que estén al alcance de casi todo el mundo.



El interruptor-transistor
Un interruptor está abierto o cerrado. En un transistor, los dos estados están representados por una corriente que fluye o no fluye. Esto se controla aplicando un voltaje al hilo de mando

Los inventores del transistor
El Premio Nobel de 1956 fue otorgado al equipo que con sus investigaciones hizo posible la invención del transistor en 1947. En la fotografía, en los laboratorios Bell Telephone (de izquierda a derecha): doctores John Bardeen, William Shockley y Walter Brittain



© Associated Press Ltd.

Preguntas y respuestas

Preguntas acerca de los ordenadores que no suelen ser contestadas en los manuales o revistas especializadas

¿Los ordenadores pueden tener emociones?
Por el momento, los ordenadores no tienen, ni pueden tener, emociones. En realidad, la pregunta verdaderamente interesante sería: ¿Por qué no? Los ordenadores actualmente en el mercado no son inteligentes —no pueden pensar por sí mismos—. No se sabe cuánto tiempo se tardará en producir ordenadores que piensen, pero probablemente podremos verlo. Algunos investigadores afirman que los procesos de pensamiento creativo son inseparables de las emociones. Así pues, según esta teoría, los ordenadores que puedan pensar, serán ordenadores con emociones.



¿Cuál es la diferencia entre un ordenador y un robot?
En realidad, los robots son prolongaciones mecánicas de los ordenadores; los brazos y los ojos hacen lo que el "cerebro" del ordenador les transmite. Los robots que actualmente ayudan en la fabricación de coches y sistemas estéreo, contienen microordenadores, pero todavía son bastante "torpes". Si se les coloca frente a una situación nueva e inesperada, simplemente no saben qué hacer. Pero los robots del mañana incluirán ordenadores más sofisticados, y los robots con inteligencia limitada puede decirse que están ya a la vuelta de la esquina.



¿Por qué son tan caros algunos tipos de software? Además, los programas profesionales frecuentemente cuestan muchísimo más que los programas de juegos.
El escribir programas extensos, sobre todo software comercial, que es verificado cuidadosamente, requiere equipos de programadores bien retribuidos trabajando durante meses o años. Para recuperar las enormes inversiones financieras y obtener un beneficio, las compañías que realizan software tienen que vender sus productos a precios que garanticen el cubrir los costos. De un ordenador para juegos se pueden vender cientos o incluso miles de unidades, por tanto un precio al por menor no muy elevado puede asegurar un beneficio. Si el mercado potencial está estrictamente limitado, el precio al por menor deberá ser más alto. Muchos programas son altamente especializados; por ejemplo, un paquete de programas para presupuestos de una imprenta tiene un mercado potencial limitado al número de imprentas del país. Una inversión de miles o millones de pesetas deberá recuperarse tanto si las ventas son de decenas o centenas como de millares de ejemplares.

Se dice que el chip de silicio dejará sin trabajo a millones de personas. ¿Cómo puede un microordenador crear problemas de desempleo?
Los efectos sociales a largo plazo creados por el microordenador son de difícil predicción, pero lo que parece evidente es que estamos asistiendo al principio de una nueva revolución industrial. Los ordenadores, en particular los miniaturizados y los micros de bajo coste, conectados con robots mecánicos, pueden ser adaptados fácilmente para reemplazar mano de obra cara. Ni los trabajos cualificados están seguros. La teneduría de libros y la contabilidad pueden efectuarse mediante programas de ordenador. También están amenazados los puestos de trabajo de los tipógrafos de los periódicos, pues los procesadores de textos de los periodistas pueden conectarse directamente al equipo de composición electrónica. Los ordenadores pueden efectuar procesos aritméticos complejos tan rápidamente, y los robots realizar operaciones mecánicas de manera tan eficiente, que cada vez son necesarios menos empleados para un trabajo.



¿Los ordenadores pueden usarse para robar un banco o desencadenar la tercera guerra mundial?
Puesto que los ordenadores pueden comunicarse entre sí utilizando las líneas telefónicas ordinarias, en teoría sería posible intervenir el ordenador central de un banco y transmitir órdenes para transferir fondos de una cuenta a otra. En la práctica las cosas no son tan sencillas. Los bancos usan métodos avanzados de protección de datos para asegurar que nadie no autorizado tenga acceso a la información confidencial. Las técnicas empleadas comprenden métodos secretos para codificar toda la información. Estos códigos son casi imposible de descifrar y en muchos casos los empleados del banco no tienen acceso a ellos. Uno de los códigos usados para información altamente confidencial es tan difícil de descifrar que se ha estimado que el ordenador más potente del mundo tardaría miles de millones de años en conseguirlo.



David Higham

Interferir un sistema de ordenadores militar sería aún más difícil. Generalmente, los ordenadores militares no utilizan las líneas telefónicas públicas. Las conexiones por microondas o vía satélite no son fácilmente accesibles para las personas corrientes. Incluso en el caso de que fuera posible interceptar una conexión por microondas portadora de información de un ordenador militar, todavía persistiría el problema nada sencillo de descifrar el código.

Commodore 64

El último ordenador personal Commodore ofrece gráficos móviles "sprite", y utiliza su TV o altavoces de hi-fi para generar un sonido de alta calidad

El Commodore 64 es realmente el primero de una nueva generación de ordenadores personales, adecuado tanto para juegos como para ayuda en la gestión de pequeños negocios.

Una memoria estándar de 64 Kbytes es suficiente para permitirle elaborar gráficos sofisticados, o programas profesionales tales como hojas electrónicas, tratamiento de textos y bases de datos. Cierta compatibilidad con los sistemas Commodore VIC-20 y PET ha incrementado la gama de software disponible en el mercado.

La gama de interfaces del 64 le permite funcionar con la mayoría de tipos de periféricos, incluso la unidad de disco y las impresoras VIC. Un sistema completo, compuesto por ordenador, unidad de disco e impresora, está al alcance de cualquier bolsillo.

Dos de las características más importantes del Commodore 64 son: gráficos "sprite" (véase p. 45) y síntesis musical total. Los "sprites" son objetos visuales creados en la pantalla mediante gráficos de alta resolución, que pueden ser movidos con una orden simple —ideal para invasores del espacio, aviones, explosiones, etc.—. Tales efectos son posibles sin gráficos "sprite", pero requieren mucha más programación. En el 64, es posible mover los "sprites", hacer que aumenten de tamaño, disminuyan o cambien de color, o que pasen por delante o por detrás de otros "sprites" u objetos gráficos estacionarios, por ejemplo un escenario de fondo —dando profundidad a la escena—. También es posible detectar cuándo dos "sprites" han chocado (¡indicación para una explosión!).

La sintetización del sonido es igualmente sofisticada, en contraste con los simples *beeps* y *sqwawks* de máquinas más baratas. Además de tener tres voces separadas (que permiten acordes y armonías en vez de sólo notas simples), el Commodore 64 permite un control completo sobre los diversos parámetros que gobiernan el sonido o el timbre de la nota que se emite. En otras palabras, el 64 puede simular una variedad completa de instrumentos musicales, y otros ruidos abstractos.

El punto débil del modelo 64 es su lenguaje BASIC, que prácticamente es la misma versión que Commodore utilizaba en sus primeros ordenadores. En vez de tener una amplia gama de órdenes prácticas para obtener un buen rendimiento de sus características, por lo demás espléndidas, la mayoría de las operaciones sofisticadas requieren la incómoda orden POKE. Afortunadamente, en la actualidad ya se pueden comprar cartuchos suplemento para remediar esta deficiencia.



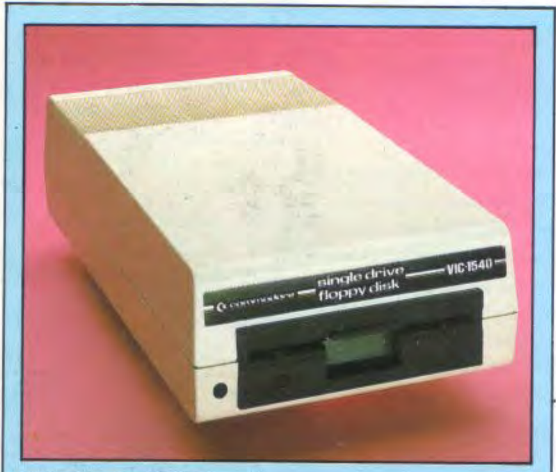
Teclado del Commodore 64

El teclado del Commodore 64 es excelente, con teclas "esculpidas" (contorneadas para facilitar su manejo). Además de los caracteres normales, tiene una amplia gama de caracteres gráficos. Una tecla multiuso grabada con el logotipo de Commodore cambia los diferentes juegos de caracteres. Los colores cambian con la tecla CONTROL (CTRL) y una de la fila superior. Las cuatro teclas-función, situadas a la derecha del teclado, pueden desempeñar cometidos especiales dentro de un programa, y por tanto permiten la entrada rápida de órdenes especiales.





Cassette 1530 C2N
Ésta es la unidad cassette estándar del equipo Commodore, y está diseñada para enchufarla directamente a la conexión para cassette del 64. Toma la corriente del ordenador, el cual puede también conectar o desconectar el motor de la cassette



Unidad de disco 1540
Se adapta al ordenador 64 a través de la conexión serial. Un disco de 5 ¼ pulgadas suministra 170 Kbytes de almacenamiento, y el tiempo máximo necesario para acceder a cualquier información almacenada es de 2 segundos



Impresora 1525
Es capaz de imprimir todos los caracteres que pueden aparecer en la pantalla del 64, incluso los gráficos. Se enchufa a la conexión serial. Imprime 30 caracteres por segundo

Conexión dispositivos
Con un enchufe de 24 patillas, se pueden conectar diversos dispositivos al ordenador. Varias patillas pueden ajustarse como líneas de entrada o salida

Conexión teclado
En este punto se conecta el teclado al microprocesador

RAM
Ocho chips suministran los 64 Kbytes del RAM de la máquina. No todos los bytes están disponibles para la programación, ya que en cuanto se conecta el ordenador, éste se carga con el programa BASIC y otras rutinas del sistema

Microprocesador
El 6510 es el procesador principal. Controla todas las operaciones y maneja buena parte de las entradas/salidas del sistema

Chip de sonido
El dispositivo interface 6581 genera todos los sonidos que puede producir el Commodore 64



COMMODORE 64

MEDIDAS
404 × 216 × 75 mm

PESO
1 820 g

VELOCIDAD RELOJ
1 MHz

MEMORIA
Un total de 64 Kbytes. 20 Kbytes de ROM suministran el sistema operativo y el lenguaje BASIC. El usuario puede disponer de un máximo de 54 Kbytes de RAM si no se emplea BASIC

VISUALIZACION EN VIDEO
25 líneas de 40 caracteres. Alta resolución con 320 × 200 puntos. 16 colores

INTERFACES
Conexión cassette, TV, conexión cartuchos, conexión monitor, interface RS232, conexión dispositivos

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
FORTH, COMAL, PILOT, LOGO, UCSD, PASCAL y otras versiones de BASIC

VIENE CON
Unidad de alimentación eléctrica, cable antena, manual

TECLADO
Teclas estilo máquina de escribir, disposición QWERTY con 62 teclas y otras cuatro para funciones

DOCUMENTACION
La Guía del Usuario (de bajo nivel) junto con los manuales (Commodore). Se describen la máquina, su funcionamiento y lenguaje, pero no a un nivel adecuado para principiantes. No suministra una guía amplia del tratamiento de los registros especiales. Esto es particularmente desafortunado puesto que el sonido y los gráficos del 64 se programan mediante estos registros especiales

Puntuación

Por qué al escribir un programa de ordenador se debe prestar atención a cada detalle de puntuación

Quizá haya advertido, en la relación del programa de la primera parte de programación BASIC (p. 20), que al final de la línea 50 hay un punto y coma. La función de este signo de puntuación en BASIC no se explicó en ese momento; sin embargo, es muy importante. El punto y coma se emplea en casi todas las versiones de BASIC para indicar que las próximas PRINT continuarán imprimiendo en la misma línea y al lado del último carácter impreso. Las líneas 50 y 60 de la página 20 eran:

```
50 PRINT "CREO QUE EL NUMERO DIGITADO ERA";
60 PRINT A
```

La línea 50 imprime las palabras entre comillas. La línea 60 imprime el valor de la variable A. El poner el punto y coma hace que el valor de la variable A sea impreso directamente tras las palabras entrecomilladas de la línea 50. Si no se utilizara el punto y coma, el valor de la variable A sería impreso en la línea siguiente a la de las palabras.

El programa que aparece a continuación está concebido para ilustrar algunas de las propiedades útiles del punto y coma, tal como se emplea en BASIC. Intente escribir el programa y hacerlo funcionar. A partir de ahora, omitiremos la advertencia <CR> al final de cada línea para indicar que debe apretar la tecla RETURN. Este programa le permite introducir una gama de temperaturas en grados centígrados (llamados también Celsius) y obtener automáticamente su equivalente en grados Fahrenheit.

```
10  REM PROGRAMA PARA CONVERTIR GRADOS C A F
20  PRINT "INTRODUCIR LA TEMPERATURA MAS BAJA"
30  INPUT L
40  PRINT "INTRODUCIR LA TEMPERATURA MAS ALTA"
50  INPUT H
60  FOR X = L TO H
70  LET F = X * 9/5 + 32
80  PRINT X; "EN CENTIGRADOS ES" F;
   "EN FAHRENHEIT"
90  NEXT X
100 END
```

Introduzca este programa, apriete LIST para verificar que ha sido incorporado correctamente y luego apriete RUN. Primero se le pedirá que introduzca la temperatura más baja. Teclee −5. A continuación deberá incorporar la temperatura más alta. Teclee 10. Y ahora el programa convertirá todas las temperaturas, en intervalos de un grado, desde −5 a 10 en unidades centígradas a sus equivalentes en la escala Fahrenheit. En la pantalla deberá aparecer algo semejante a:

```
-6   EN CENTIGRADOS ES 21,2   EN FAHRENHEIT
-5   EN CENTIGRADOS ES 23   EN FAHRENHEIT
-4   EN CENTIGRADOS ES 24.8   EN FAHRENHEIT
-3   EN CENTIGRADOS ES 26,6   EN FAHRENHEIT
-2   EN CENTIGRADOS ES 28,4   EN FAHRENHEIT
-1   EN CENTIGRADOS ES 30,2   EN FAHRENHEIT
-0   EN CENTIGRADOS ES 32   EN FAHRENHEIT
```

Observe que las columnas no son exactamente iguales debido a la coma de los decimales, pero cada valor en grados centígrados está impreso con su equivalente en grados Fahrenheit en una sola línea. Después de pasar el programa unas cuantas veces, vuelva a escribir la línea 80 tal como está, pero sustituyendo todos los punto y comas por comas. Pase el programa otra vez. Como puede apreciar, ahora aparece todo desordenado.

Para comprobar por qué ha sucedido esto, probemos un programa muy simple, con el fin de comparar el efecto de las comas con el de los punto y comas. Digite NEW <CR>. Introduzca:

```
10 REM COMPARAR; CON,
20 PRINT "ESTA LINEA USA PUNTO Y COMAS"
30 PRINT "H"; "E"; "L"; "P"
40 PRINT "ESTA LINEA USA COMAS"
50 PRINT "H", "E", "L", "P"
60 END
```

Cuando el BASIC imprima la línea 30, aparecerá en la pantalla HELP; en cambio en la línea 50 será H E L P. (Véase el recuadro "Complementos al BASIC" para las variaciones entre ordenadores diferentes.) La coma tiene muchos usos en el BASIC, pero con relación a PRINT tiene el efecto de que cada letra aparezca en la pantalla (o impresa en papel) espaciada, normalmente entre 8 y 16 espacios, según la versión del BASIC. Si PRINT se utiliza sin comas o punto y comas, las letras aparecerán en líneas distintas.

Además de ilustrar sobre el uso del punto y coma en el BASIC, nuestro programa de conversión de temperaturas vuelve a examinar varios puntos de los dos primeros apartados de nuestro curso de programación BASIC. Las líneas 30 y 50 sitúan las variables L y H en los valores mínimo y máximo de las temperaturas que se quieren convertir. La línea 60 es la primera parte de un bucle FOR-NEXT. Parece diferir del bucle FOR-NEXT que hemos encontrado hasta ahora al utilizar letras en vez de números. De hecho, no hay ninguna diferencia.

Las letras que empleamos aquí, L y H, son variables con valores numéricos que corresponden a los valores digitados en el INPUT L y el INPUT H. Si, como se ha sugerido anteriormente, ha introducido −5 y 10, la expresión FOR X = L TO H es equivalente a FOR X = −5 TO 10.

La línea 80 dice: PRINT el valor de X (que empieza a la temperatura más baja y cada vez se incrementa en 1 hasta la temperatura más alta), seguido directamente en la misma línea (ésta es la razón del uso del punto y coma) por las palabras entrecomilladas, seguidas a su vez directamente (otro punto y coma) por el valor de F. Si se fija detenidamente en F, comprobará que es el valor de la temperatura en grados centígrafos, convertidos en Fahrenheit al multiplicar por nueve, dividir por 5 y sumarle 32. La línea NEXT X asegura la continuación del proceso hasta alcanzar el límite superior en el bucle FOR-NEXT.

Antes de continuar con una variación más sofisticada en el apartado PRINT, merece la pena revisar la línea 70 de nuestro programa de conversión de temperaturas:

```
70 LET F = X* 9 / 5 + 32
```

Esta línea asigna un valor a la variable F (abreviatura de Fahrenheit). El programa, primero, toma el valor de X (temperatura en centígrados), lo multiplica por 9, lo divide por 5 y le suma 32. La expresión de esta fórmula en un libro corriente de física sería F = C × 9 : 5 + 32. El lenguaje BASIC utiliza los símbolos * para la multiplicación, / para la división, + para la adición y − para la sustracción.

En la aritmética corriente, y también en BASIC, es importante el orden en que se efectúan las operaciones. La multiplicación tiene siempre la máxima prioridad, seguida de la división, la adición y la sustracción. Si una parte de la expresión matemática está entre paréntesis, debe calcularse en primer lugar. Si se quiere que una adición sea efectuada antes que una multiplicación, debe ponerse entre paréntesis. Por ejemplo, si quiere saber el saldo de su cuenta corriente en dólares más sus ahorros en la misma moneda, podría expresarlo en parte de un programa semejante al siguiente:

```
D = (C + S)* 0,0066
```

Si su cuenta corriente es de 120 000 pesetas (C) y sus ahorros son de 300 000 pesetas (S) y una peseta vale 0,0066 dólares, primero se sumarán las pesetas (C + S) y luego se multiplicará por 0,0066 para convertirlas en dólares. Sin el paréntesis, el valor de sus ahorros sería multiplicado en primer lugar por 0,0066 y luego sería añadido al resultado el valor de la cuenta corriente. ¡Que no era lo que quería calcular! Compruebe siempre que las operaciones aritméticas sean realizadas en el orden correcto.

Print Using

Para comprobar una última característica importante de nuestro programa de conversión de temperaturas, digítelo otra vez y póngalo en funcionamiento. Introduzca −10 como valor de la temperatura más baja y 10 para la más alta. Como ya hemos visto, la impresión en la pantalla es muy desigual. Esto es debido a los punto y comas utilizados en la línea 80 para concatenar (unir) todas las partes que se imprimen, en vez de hacerlo en líneas separadas. Lo cual es positivo, excepto que el espacio ocupado por las cifras —en grados centígrados y Fahrenheit— varía. Esto hace que las columnas queden desalineadas.

Casi todas las versiones de BASIC tienen una característica PRINT llamada PRINT USING, que permite poner en orden la impresión de las palabras o de los números. Si se quiere imprimir el valor de X y se sabe por adelantado que varía desde —99 a 99, las cifras pueden imprimirse correctamente alineadas usando PRINT USING "# # #"; X. Los tres signos permiten imprimir hasta tres dígitos, o dos dígitos precedidos del signo menos. Si se introducen más de tres dígitos, no serán impresos correctamente. Sin embargo, si se introducen sólo uno o dos dígitos, serán colocados correctamente. Si son necesarias comas decimales, pueden incluirse en la posición correcta con los signos. Por ejemplo, la expresión puede tener la forma PRINT USING "# # # · # #"; X. Utilice un signo para cada dígito. Todas las comas decimales se alinearán automáticamente.

Modifique el programa original, cambiando la línea 80 y añadiendo las líneas 82, 84 y 86:

```
80 PRINT USING "# # #"; X;
82 PRINT "EN CENTÍGRADOS ES";
84 PRINT USING "# # # · # #"; F;
86 PRINT "EN FAHRENHEIT"
```

Digite otra vez LIST y RUN. Todas las columnas aparecerán ahora perfectamente alineadas.

En los próximos capítulos del curso, veremos cómo se puede "guardar" un programa, no siendo necesario volverlo a escribir cada vez.

Ejercicios

- Introducir una "temperatura más baja" que −1 000. ¿Por qué no funciona ahora el programa? ¿Cómo modificaría el PRINT USING en la línea 80 para hacerlo funcionar?
- Cambiar la línea 84 de forma que sólo se impriman números enteros (sin decimales).
- Escribir un programa para convertir una serie de cifras de pesetas a dólares, utilizando un valor de cambio de 0,0066 dólares por peseta.

Complementos al BASIC



Este mando no existe en el Commodore 64, Oric, Spectrum, ZX81 o BBC Micro. Sin embargo, el BBC puede limitar el número de decimales a imprimir, lo cual se consigue utilizando la siguiente instrucción: @% = 131594



El uso de la coma entre campos impresos separará los temas a imprimir insertando un número de espacios, que varían según el ordenador que se emplee. En el BBC y Commodore 64, son 10 espacios; en el Dragon, Spectrum y ZX81 son 16. El Oric utiliza sólo 4 espacios

Cuando 1 y 1 son 10

Los ordenadores realizan sus prodigiosos cálculos únicamente con dos dígitos: 0 y 1

Conversión al sistema binario



La forma más fácil de convertir números binarios pequeños a sus equivalentes decimales consiste en imaginar que se escribe el "valor de posición" de cada columna binaria en los dedos de la mano derecha. Siempre que el número binario no exceda de cuatro dígitos, todo lo que hay que hacer es extender el dedo adecuado si el dígito binario correspondiente es un 1, o doblarlo si es 0.
Cuando se extienden los dedos adecuados para el número 1010, se obtiene 8 y 2, que sumados dan el número 10 en el sistema métrico decimal. La tercera ilustración muestra cómo decodificar 0101: da un 4 y un 1, que se convierten en 5 en la forma decimal. Mediante este método, intente hallar los equivalentes decimales de los números 1110 y 0110.
El método puede ampliarse utilizando las dos manos para expresar números binarios mayores. Al objeto de realizar esta operación, los dedos de la mano izquierda (con la palma frente a usted) deben corresponderse con los valores 16, 32, 64 y 128, con el valor 16 para el dedo meñique y el 128 para el índice

Elaine Keenan

Mucha gente está tan acostumbrada a nuestro sistema numérico que no se le ocurre pensar que puedan existir otros.

Los romanos idearon un sistema para la representación numérica utilizando las letras del alfabeto. Así, la X significaba 10, la L 50, la C 100, la D 500, etc. El sistema romano resultaba idóneo como forma de registrar números simples; sin embargo, no sería apto para el cálculo con ordenador. En números romanos, incluso las sumas son difíciles, por una razón: no existe el concepto de "valor de posición"; la posición de un dígito no aporta ningún dato sobre cuál es su "valor".

Fíjese en los números 506 y 56. La única diferencia aparente es el cero que aparece en medio del primer número. Su función consiste en informarnos que en el número 506 no hay "decenas", sólo cinco "centenas" y seis "unidades".

Cada "columna" o posición en un número convencional tiene un "valor" asociado con él. La columna de la derecha del número es la de las "unidades", la siguiente (en dirección a la izquierda) es la columna de las "decenas", la próxima la de las centenas y así sucesivamente. El dígito utilizado en cada "columna" significa qué valor de ella está representado.

Usted puede preguntarse qué relación tiene lo expuesto anteriormente con los ordenadores y el sistema binario. Los ordenadores son máquinas electrónicas, que pueden operar fácilmente con números mediante la utilización de niveles de voltaje. Cinco voltios representan un uno, y cero voltios un cero. Como aprendimos en "Bits y bytes" (p. 28), los unos y ceros resultan perfectamente adecuados para representar cualquier número, por grande que sea.

Usando el conocido sistema decimal de base 10 (denominado también *sistema denario*), el número 506 es una manera concisa de representar el equivalente a

quinientos seis nudos en una cuerda o quinientas seis muescas en un bastón. En aritmética binaria, este número se representa por un incómodo 111111010.

Ya que el sistema empleado aquí es el "binario", el valor de situación de cada dígito y de cada columna es distinto. En vez de incrementar el valor en potencias de 10, las columnas aumentan en potencias de 2.

La columna de la derecha sigue siendo la de las "unidades", pero como sólo hay dos símbolos (0 y 1), nos quedamos sin dígitos en cuanto empleamos el 1. En el sistema decimal, esta circunstancia sólo se presenta cuando llegamos al 9; la próxima columna usa un dígito que dice: nos hemos quedado sin símbolos —no tenemos nada para representar números mayores que 9—, por esta razón utilizaremos la columna de las "decenas" y con un 1 indicaremos que ahora tenemos un "conjunto" de diez.

El sistema binario funciona de la misma forma. En vez de formar grupos de diez y escribir 10, el sistema binario agrupa conjuntos de 2, y por tanto los dígitos binarios 10 representan el número decimal 2.

Si escribimos el número quinientos seis en los sistemas decimal y binario, advertiremos claramente la semejanza esencial existente:

Centenas	Decenas	Unidades	
5	0	6	
= 5 × 100 +	0 × 10 +	6 × 1	(= 506)

256	128	64	32	16	8	4	2	1
1	1	1	1	1	1	0	1	0

= 1×256 + 1×128 + 1×64 + 1×32 + 1×16 +
+ 1×8 + 0×4 + 1×2 + 0×1 (= 506)

Las reglas aritméticas en el sistema binario son exactamente las mismas que las del conocido sistema decimal: la única diferencia consiste en que después del 1 ya no hay otros símbolos numéricos, en lugar de suceder esto con el 9. Realicemos unas sumas para probarlo. Los equivalentes decimales están representados entre paréntesis.

(3)	11
+(5)	+101
(8)	1000

(1 + 1 = 0 se acumula 1)
(1 (acumulado) + 1 = 0 se acumula 1)
(1 (acumulado) + 1 = 0 se acumula 1)
(1 (acumulado) + 0 = 1)

En el sistema binario, como se observa, sumar 1 y 1 significa que se agotan los símbolos, puesto que sólo se pueden emplear unos y ceros. Por tanto se dice: "Uno y uno igual a cero y se lleva uno" (al igual que en el sistema decimal, en el que si sumamos 1 y 9 agotamos los símbolos —no existe ningún símbolo mayor que 9—, y por ello decimos de nuevo: "Nueve y uno igual a cero y se lleva uno"). A continuación se representa una suma ya resuelta y otras dos que usted mismo debe solucionar.

(4)	100	(7)	111	(3)	11
+(6)	+110	+(2)	+ 10	+(12)	+1100
(10)	1010	(?)	?	(?)	?

Después de lo expuesto, habrá observado que los números binarios son mucho más largos que sus equivalentes en el sistema decimal. Intente sumar 11010110 a 1101101 (recuerde que debe mantener las columnas alineadas del mismo modo que si realizara una suma en el sistema decimal de un número con otro más pequeño).

Historia de los números



Babilónico **Romano** **Hindú** **Binario**

Los antiguos babilonios tenían un sistema numérico avanzado, basado en el número 60 en vez del 10. En la ilustración se representa el número 59 en escritura cuneiforme babilónica. El empleo del 60 como número base tenía muchas ventajas e incluso en la actualidad quedan algunas reminiscencias de este sistema. Un minuto tiene 60 segundos, una hora 60 minutos y seis veces 60 grados en un círculo: todo ello constituye un vestigio de aquel sistema matemático perfeccionado hace 4 000 años.

El sistema romano representó un atraso considerable. Los números se representaban con las letras del alfabeto, pero la posición de cada cifra no indicaba su valor, por lo que resultaba prácticamente imposible realizar la operación aritmética más simple.

Los hindúes utilizaban nueve signos para designar los números del 1 al 9 y más tarde añadieron otro signo que representaba el cero. Pero su contribución vital fue la introducción del "valor de posición": la idea de que la posición de un dígito en un número determina su "valor". Así el "valor" de 3 en 30 es de tres decenas. Los árabes adoptaron el sistema hindú, que gradualmente se extendió por Europa. Uno de los matemáticos árabes más importantes se llamaba Al-Jwarizmi. La pronunciación latinizada de su nombre determina el término matemático de algoritmo, y su libro *Al-yabr wa'l Mugabalah* nos trae a la memoria la palabra álgebra.

Los ordenadores utilizan el sistema binario porque los números, con independencia de su magnitud, pueden representarse usando sólo unos y ceros

Andy Leslie

Mandos veloces

Algunos dispositivos que ayudarán a los entusiastas de los juegos por ordenador a acelerar la acción



En un juego por ordenador, quizá tenga que pilotar una nave espacial a través de las líneas enemigas y disparar sus misiles para destruir un blanco. La palanca de mando transfiere el control de la nave espacial desde un teclado sofisticado hasta sus propias manos. Es una réplica fiel de las que llevan los aeroplanos.

La palanca de mando se conecta a la parte trasera del microordenador. La nave espacial, o cualquier otro objeto que se controle, se mueve en la misma dirección que la palanca. Normalmente puede desplazarse en cualquiera de los cuatro sentidos. Cuando la palanca se desplaza hacia adelante, la nave se mueve hacia la parte superior de la pantalla. Eléctricamente, en el interior del mecanismo hay cuatro conmutadores colocados de tal forma que cuando se mueve la palanca, uno, y sólo uno, de los contactos está cerrado: Cada uno de los conmutadores envía su propio mensaje al ordenador para desplazarse en diferente dirección: arriba, abajo, derecha o izquierda.

Algunas palancas de mando tienen también un botón para disparar los misiles. El botón está a un lado de la palanca y debe ser accionado con la otra mano. O bien, en la empuñadura de la palanca, los misiles son lanzados apretando el dedo pulgar.

Los microordenadores más baratos, en particular el Sinclair ZX81 y el Spectrum, no siempre ofrecen la posibilidad de conectar una palanca de mando. Por tanto se deberá o bien apretar la tecla de la dirección en que se desea que se desplace, o bien comprar una interface para la palanca.

La interface es un adaptador que permite conectar una palanca al ordenador. Algunas compañías independientes han fabricado interfaces para estas máquinas, pero incluso con este dispositivo es necesario escribir el programa del juego para introducir el control de la palanca y el del teclado.

Palanca de mando

Pulsador de disparo
Se utiliza para el lanzamiento de los misiles o para hacer fuego con el rayo láser. En otros programas, el pulsador tiene la ventaja de efectuar el control con un solo mando

Potenciómetros
Son utilizados frecuentemente en dispositivos electrónicos en los que se tenga que variar el voltaje. El control del volumen o del tono en un aparato de hi-fi se realiza de la misma forma. Los potenciómetros tienen una serie de resistencias eléctricas y un contacto deslizante. El valor de la resistencia en el circuito cambia al desplazar el contacto. El ordenador mide el cambio de resistencia y traduce esta información en un movimiento del cursor en la pantalla. Un potenciómetro controla el movimiento vertical y el otro el horizontal

Apoyos basculantes
La palanca está sostenida por estos dos apoyos, montados perpendicularmente entre sí. Están conectados a los potenciómetros. Cuando se mueve la palanca, los contactos de los potenciómetros se deslizan y cambia la resistencia eléctrica

David Weeks

Pulsador de disparo
Al apretarlo, los dos contactos se unen y el conmutador queda cerrado. Al dejar de presionar el pulsador, los contactos se separan por la acción de un resorte

Bola
Al presionarla, los dos contactos se unen y el conmutador queda cerrado. Al soltarla, los contactos se separan por la acción de un resorte

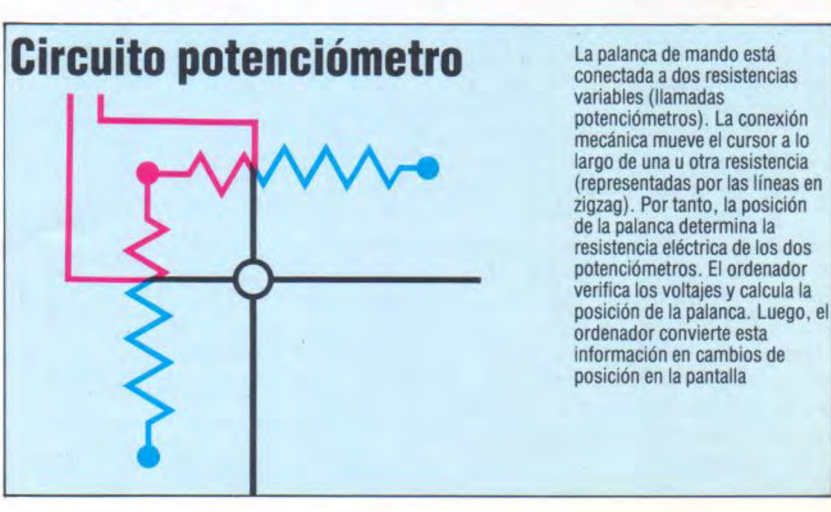
Luz y fotocélula
Al girar el disco ranurado se interrumpe el rayo luminoso y el ordenador cuenta el número de destellos detectados por la fotocélula. El cursor u objeto se desplaza a través de la pantalla en proporción a este número. Existen dos juegos de disco y fotocélula para controlar tanto el movimiento horizontal como el vertical en la pantalla

Rodillo
El contacto de la esfera hace girar al rodillo, y éste guía el disco ranurado

Bola de mando

Imagínese que está guiando el cursor de la pantalla a través de un laberinto. Tiene que ser capaz de hacer avanzar el cursor y de guiarlo a través de los pasajes a medida que se entrelazan y giran. La bola de mando está diseñada para este tipo de problemas. Es una esfera del tamaño de una bola de billar, que se hace girar con la palma de la mano. Al girar la bola, el objeto se mueve en la misma dirección, obteniéndose un control completo e inmediato. En el interior del aparato hay dos juegos de ruedas situadas perpendicularmente entre sí, que rozan con la bola. Al hacer girar ésta con la palma de la mano, una de las ruedas capta la parte vertical del movimiento y la otra la horizontal. El ordenador debe unir ambas señales para conseguir la trayectoria.

Circuito potenciómetro



La palanca de mando está conectada a dos resistencias variables (llamadas potenciómetros). La conexión mecánica mueve el cursor a lo largo de una u otra resistencia (representadas por las líneas en zigzag). Por tanto, la posición de la palanca determina la resistencia eléctrica de los dos potenciómetros. El ordenador verifica los voltajes y calcula la posición de la palanca. Luego, el ordenador convierte esta información en cambios de posición en la pantalla

Reenvío total

Cómo "vigila" el ordenador toda la información almacenada en su memoria, asegurándose de que nunca se olvide y tenga un acceso inmediato

En términos humanos, la memoria es el almacén de la mente, el lugar donde se acumulan los detalles de la experiencia para su uso posterior. Y en términos de ordenador, "memoria" significa prácticamente lo mismo, sólo que la memoria de un ordenador tiene un campo de acción más limitado.

Para un ser humano, tener poca memoria es un inconveniente. Para un ordenador, es un desastre. Sin memoria, el ordenador no tendría nada en qué basarse ni nada que le indicara lo que tiene que hacer, ya que también utiliza su memoria para almacenar los programas que lo guían.

En ambos casos, la palabra "memoria" implica dos cosas: almacenaje y reenvío de información. Almacenar datos sin la posibilidad de volverlos a extraer, no es muy útil, e intentar obtener información que no ha sido almacenada es obviamente inútil.

Los dos tipos de memoria son también similares desde otro punto de vista. Parece ser que la memoria humana es de dos tipos generales: de corta y de larga duración. Un hombre que debe cruzar una carretera, por ejemplo, recordará que debe esperar hasta que pase el vehículo que se acerca. Pero cuando esté en el otro lado de la carretera, se olvidará totalmente del automóvil. Su memoria del coche era, pues, de corta duración.

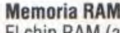
Sin embargo, si el mismo coche hubiera estado ocupado por dos hombres encapuchados, sentados en el asiento trasero, y fuera conducido por la mujer del hombre que cruza, éste podría recordar bien todo el incidente, ¡incluso el modelo y color del coche, y posiblemente el número de matrícula! Esto es memoria de larga duración.

Se puede decir que también los ordenadores tienen memoria de corta y larga duración. La de larga duración o "no volátil" contiene los programas e información que el usuario quiere guardar. Estos programas se almacenan como grabaciones magnéticas en la superficie de una cinta de cassette, discos flexibles o en la ROM.

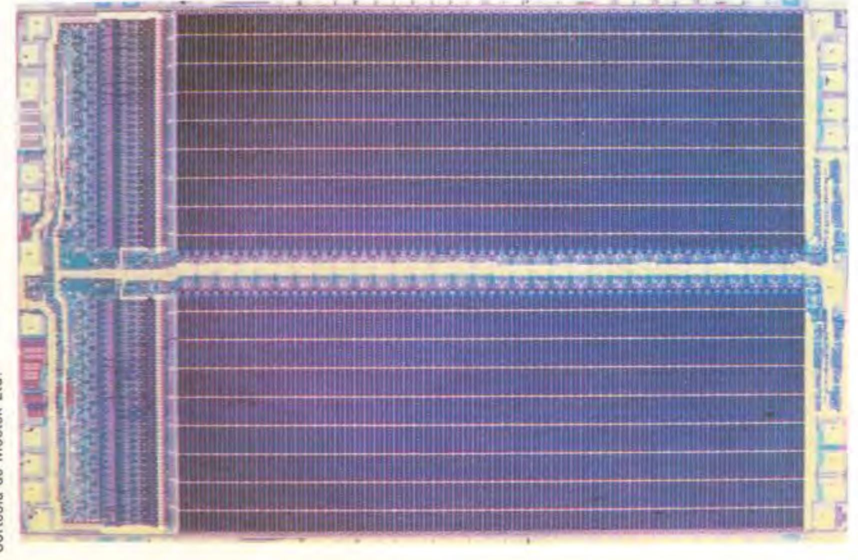
La memoria de corta duración o "volátil" es el chip RAM, situado en el interior del mismo ordenador, y sólo se utiliza temporalmente mientras el ordenador está trabajando. En el momento en que se corta el suministro de energía, aunque sea durante una fracción de segundo, todo el contenido de esta memoria desaparece instantáneamente.

Sin embargo, la mentada analogía con la memoria humana no es totalmente exacta. Para que el ordenador funcione, es necesario traspasar los programas y datos adecuados desde el almacenamiento de larga duración al de corta duración, para que así el ordenador pueda tener un acceso instantáneo a ellos. Y la *forma* en que los datos son almacenados y reenviados desde o hacia la memoria del ordenador es completamente diferente al proceso que se produce en los seres humanos.

La forma en que funciona la memoria humana es aún un misterio, puesto que los recuerdos de un incidente determinado no parece ser que se almacenen en un pequeño segmento identificable del cerebro. No tenemos que descifrar dónde está situado un tema determinado y reenviarlo al primer plano de la mente. Y cuando hemos terminado de recordar, no tenemos que preocuparnos de volverlo a situar en un punto determinado del cerebro.



Memoria RAM
El chip RAM (abajo) es uno de los adelantos recientes más importantes en la tecnología del ordenador. La RAM (Random Access Memory: memoria de acceso directo) es una de las variedades de la memoria totalmente electrónica, una categoría que incluye también la ROM (Read Only Memory: memoria de lectura solamente). Las cintas de cassette y los discos flexibles magnéticos son ejemplos de otro tipo de memoria: la electromagnética. La memoria RAM se fabrica con silicio, usando un proceso fotográfico y reacciones químicas, para crear miles de diminutos transistores. Cada "bit" de memoria necesita como mínimo un transistor en el circuito.
El tiempo requerido para "escribir" un simple bit en una de las 16 384 celdas de almacenamiento es de unos 200 nanosegundos (una quinta millonésima parte de segundo)



Cortesía de Mostek Ltd.

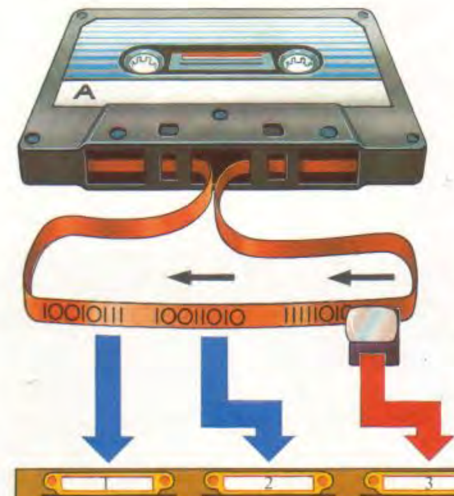
La memoria del ordenador: un caos organizado

En la memoria de un ordenador, lo que es vital es la *situación* de cada tema. El ordenador debe ser capaz de encontrar un byte de información determinado, ya sea parte de un programa o parte de los datos del mismo. El ordenador necesita también "tomar nota" de dónde pone la información.

La memoria humana se parece más a un buzón completamente lleno de información, pero desordenado. Los componentes de una información son introducidos sin orden, aparentemente al azar, mezclándose entre sí y empujados al interior del cerebro al reunir más y más imágenes y experiencia. De algún modo, el cerebro le da sentido y extrae lo que necesita, cuando lo necesita.

La memoria de un ordenador se parece más a un gigantesco casillero, estando cada casilla completamente separada de las otras. Todo está muy ordena-

Memoria



Los programas grabados en una cinta de cassette se almacenan "secuencialmente", con cada bit que compone un byte grabado uno tras otro. Al reproducir la cinta, el ordenador "lee" cada bit, pero los almacena en grupos de ocho (bytes) en cada celda de la memoria. El primer byte de la cinta es colocado en la primera celda disponible, el segundo byte en la próxima, y así sucesivamente. Cuando el ordenador necesita poner en marcha el programa, lo único que debe saber es la "dirección de comienzo".
El ordenador transfiere de forma secuencial el contenido de cada celda al interior de la unidad central de procesamiento, y los bytes hacen que ésta "ejecute" o realice las acciones pedidas por el programa.
Una parte de la memoria está ocupada por los programas del sistema responsables de aspectos fundamentales de su funcionamiento (verificación de qué teclas han sido pulsadas, colocación de los caracteres en la pantalla, etc.). Este software incorporado puede incluir también el lenguaje de programación BASIC. Estos programas internos ocupan espacio en la memoria y, por tanto, dejan menos sitio para el almacenamiento del software comercial o del usuario. Algunas versiones de BASIC, por ejemplo, son almacenadas en 16 Kbytes de memoria. Si el ordenador está provisto de 64 Kbytes de memoria, sólo quedan disponibles 48 Kbytes para otros programas
Cuando se introduce un programa a partir de una cassette, la primera posición de la memoria disponible (vacía) no será, por consiguiente, la primera posición en la RAM. Una de las primeras obligaciones del software base (sistema operativo) es saber y recordar dónde está la primera posición de la memoria disponible para el usuario. Una vez que el programa ha sido introducido en la RAM, el software base dice: "empezar por buscar en la memoria la posición x y continuar examinando cada posición sucesiva de la memoria, introducir el contenido de cada posición en la CPU y realizar lo que ésta diga". El orden original en el que ha sido introducido el programa por el programador es el mismo en el que se ha grabado la cinta. Cuando el programa es transferido desde la cinta hasta la memoria, es colocado en el mismo orden en las celdas. Para el ordenador, el efecto es el mismo que si hubiera sido introducido a través del teclado



Tony Lodge

do; cada casilla tiene un número (llamado su "dirección") y contiene sólo un byte, ni más ni menos. Así pues, el ordenador encuentra la información por el número de la casilla, no por lo que se halla almacenado en esa casilla.

Los ordenadores no tienen inteligencia, y no pueden organizar sus memorias por sí mismos. La única razón por la que un ordenador puede almacenar algo es porque alguien ha colocado la información en la casilla correcta y en el orden adecuado, y a su debido tiempo. ¿Cómo sucede esto en un ordenador doméstico normal?

Al conectar el ordenador, generalmente aparece en la pantalla un mensaje para indicarle que está en funcionamiento. En la mayoría de los casos, también le informa que puede empezar a escribir un programa. Este mensaje, y las ayudas que le permiten empezar a programar, están almacenados en una parte de la memoria interna del ordenador; es necesario almacenarlos en la memoria de larga duración (normalmente, en un chip ROM; véase p. 9).

Esta porción de la memoria contiene los programas que verifican si se han pulsado las teclas, que "imprimen" las letras en la pantalla y realizan otros trabajos "básicos" esenciales. También contiene un programa especial que traduce las órdenes, por lo general escritas en BASIC, al lenguaje binario de unos y ceros, comprensible para el ordenador.

Cuando se conecta el ordenador, el mensaje que aparece en la pantalla suele decir: "x bytes libres", donde "x" es algo parecido a 15 797 u otro número cualquiera. Lo que representa este número es las casillas de la memoria que están libres para utilizar. Al ir pulsando las teclas, se van llenando las casillas, y aquí aparece otro punto importante relativo a la memoria del ordenador: el *orden* en que se almacena la información.

Al apretar una tecla, se envía un byte (que representa la letra digitada: véase p. 3) a la memoria para su almacenamiento. Al teclear la letra "r", por ejemplo, se coloca esta letra en una casilla de la memoria en forma binaria.

Pero ¿a qué casilla va a parar esta "r"? Pues a la primera ranura de la memoria de corta duración que esté libre. Si se piensa en un bloque de casillas vacías colgado de una pared, la "r" iría a la casilla superior de la esquina izquierda.

Si ahora se aprieta la tecla "e", la serie adecuada de bits irá a la segunda casilla vacía, a la derecha de la "r". Al teclear en tercer lugar una "d", ésta se situará en la tercera casilla junto a la "e". Al mirar el casillero, en la fila superior aparecerán los códigos de la palabra "red".

El ordenador tiene un "contador" interno para calcular qué casilla se ha ocupado. Éste sabe dónde empezar, porque el programa del sistema le dice en qué punto comienza el área libre de la memoria. Cuando se almacena una letra, la cifra del contador se incrementa en una unidad para designar la casilla siguiente.

Dinero al momento

Cómo puede obtenerse durante las 24 horas del día un servicio fiable del cajero automático de su banco

En la actualidad, gracias al microordenador, los clientes de un banco pueden tener acceso a una serie de servicios, como efectuar reintegros en dinero efectivo, solicitar un nuevo talonario de cheques, o pedir una relación detallada del estado de sus cuentas, imposibles de realizar fuera de los horarios de oficina.

En efecto, cualquiera de estas operaciones bancarias pueden llevarse a cabo siempre, o casi siempre, que los clientes lo precisen, a cualquier hora del día o de la noche, incluso en días festivos. Lo único que se necesita es una pequeña tarjeta de plástico que contiene información codificada acerca de la cuenta del cliente. Cuando se necesita dinero en un momento determinado, el cliente no tiene más que acudir a la sucursal del banco más próxima que tenga instalado un cajero automático.

Estas máquinas son en realidad microordenadores "disfrazados". Para retirar dinero, el cliente debe introducir su tarjeta en la ranura de la máquina. La tarjeta está provista de una banda magnética negra, similar al material magnético utilizado en una cinta de cassette. Cuando se inserta la pequeña tarjeta de plástico, un lector magnético verifica el código numérico impreso en la banda. La finalidad de estos números es mostrar que la tarjeta es del tipo correcto (y no una imitación) y asimismo informar al microordenador acerca de cuál es el número secreto personal del cliente.

El cajero automático pide entonces al cliente que se identifique tecleando su número secreto personal. Si el número es correcto, el cliente dispone de varias opciones, que aparecen en la pantalla de la máquina. Por lo general, estas opciones consisten en la posibilidad de retirar dinero o comprobar el saldo.

Si el cliente quiere retirar dinero, pulsa el botón adecuado y luego teclea la cantidad de dinero que necesita. En este punto, la máquina puede actuar de formas diferentes. Un código impreso en la banda magnética comunica al cajero automático si necesita consultar con el ordenador central del banco o si puede hacer la operación por sí mismo.

En este último caso, el microordenador verifica información adicional codificada en la banda magnética. Ésta le informa de la cantidad máxima de dinero que se puede retirar. Por ejemplo, si el límite es de 40 000 pesetas diarias, no se podrán retirar cantidades superiores. Entonces se procede a teclear la suma de dinero que se precisa, debiendo tener en cuenta que la máquina sólo puede entregar cantidades que sean múltiplos de mil.

Si el cliente desea realizar una consulta o no existe límite en cuanto a la cantidad de dinero que se puede retirar, el cajero automático se conecta al ordenador central del banco. Esto significa que se hace una conexión a través de líneas telefónicas especiales de forma que el microordenador puede comunicar con el ordenador central para comprobar el saldo.



El cajero automático fue introducido para prestar servicios bancarios fuera del horario de oficina. Día y noche, el cliente puede retirar dinero, pedir un nuevo talonario de cheques, comprobar el saldo o pedir una relación detallada. Al cliente se le entrega una tarjeta —semejante a una tarjeta de crédito— y se le asigna un número personal

Si en la cuenta hay saldo suficiente, el ordenador central dará instrucciones al microordenador del cajero para que el dinero pueda ser entregado. Una máquina contadora de billetes situada en el cajero cuenta el dinero solicitado y lo expulsa a través de la ranura correspondiente.

Si el cajero automático ha trabajado en conexión con el ordenador central, este último es informado de la cantidad entregada y la cuenta del cliente es automáticamente puesta al día.

Problemas con los billetes amarillos

Las tarjetas magnéticas han sido usadas incluso por los viajeros del Metro de Londres. Hace algunos años fue ideado un complejo sistema que permitía a los pasajeros, mediante la utilización de un billete magnético amarillo, entrar y salir a través de puertas automáticas. Desgraciadamente, las máquinas de la salida rechazaban los billetes con tanta frecuencia que la ira de los pasajeros obligó a abandonar el sistema



Julie-Ann Chambers



Ian McKinnell

Procesador portátil
El tratamiento de textos se está convirtiendo rápidamente en una de las más difundidas aplicaciones del ordenador. Las nuevas máquinas se diseñan con configuraciones específicas para posibilitar esta función. Estas configuraciones incluyen pantallas de 80 columnas (para visualizar la anchura total de una carta mecanografiada), unidades de disco incorporadas (en ocasiones el precio incluye un disco para tratamiento de textos) y teclas de función programables que se emplean para manipular el texto. Algunas máquinas, como la Ajile que muestra la fotografía, son portátiles, ¡ideales para periodistas y ejecutivos de alto nivel!

El texto perfecto

Con un software adecuado, su micro se transforma en procesador de textos, editando y almacenando las palabras e, incluso, corrigiendo los errores ortográficos

El tratamiento de textos es una de las tareas más útiles que pueden realizarse con un microordenador. Pero la frase "tratamiento de textos" no explica, a nivel popular, el significado de esta poderosa herramienta. Ante esta expresión, la reacción normal suele ser preguntar: "¿Cómo se pueden procesar las palabras?"

Los anuncios del periódico ofrecen a menudo "procesadores de textos para su oficina". Lo que estos anuncios olvidaban informar era que el caro hardware de estos procesadores no era otra cosa que un microordenador adaptado específicamente para realizar programas de tratamiento de textos. Los procesadores de textos son menos flexibles que los microordenadores corrientes, porque sólo pueden realizar una tarea.

Quizá la clase de programa a la que alude la frase "tratamiento de textos" debería haberse expresado como "mecanografiado auxiliado por ordenador". Con la adición de una impresora, actualmente la mayoría de los ordenadores personales pueden ejecutar algún programa de tratamiento o edición de textos. Pero es necesario que el propietario de un ordenador trate por sí mismo de procesar textos, para que comprenda lo útiles que resultan este tipo de programas.

Utilizado como procesador de textos, un ordenador visualiza las palabras en pantalla tal como éstas son digitadas, del mismo modo en que, con una máquina de escribir, éstas se imprimen sobre un papel. Los microordenadores más grandes pueden visualizar 80 caracteres en la pantalla, que representan una "página". En los ordenadores más pequeños, el usuario necesita más paciencia. Debe arreglárselas con una pantalla mucho más estrecha y, en algunos modelos, se encuentra con la falta de letras minúsculas. También ha de tener presente que una máquina pequeña sólo

puede almacenar una cantidad limitada de texto.

El programa ofrece diversos y sofisticados medios auxiliares. Todos los programas de tratamiento de textos perciben el final de cada línea a medida que éste se acerca y automáticamente "dan la vuelta", bajando completa la última palabra y colocándola al comienzo de la línea siguiente. Esto significa que el mecanógrafo ya no ha de preocuparse por "retornar el carro" al final de cada línea. Por el contrario, podrá escribir de manera fluida y continua, puesto que el programa irá creando una línea nueva cada vez que sea necesario. No obstante, cuando el texto deba proseguir en punto y aparte, el mecanógrafo tendrá que pulsar la tecla RETURN.

En una máquina de escribir convencional, usted no tendrá otra alternativa que corregir los errores mecánicamente, por lo general borrando o tachando el error y volviendo a escribir encima, lo que resulta bastante incómodo. Cuando haya una o dos correcciones, las únicas opciones serán enviar una carta emborronada o escribirla de nuevo. El tratamiento de textos resuelve el problema. El cursor intermitente que aparece en pantalla le indica, como siempre, su ubicación en cada momento. Usted deberá desplazar el cursor sobre las palabras ya escritas hasta el punto donde haya una errata. Entonces podrá hacer que el error desaparezca y volver a digitar en la forma correcta.

Una vez usted esté enterado del caudal de ayuda que puede prestarle un programa de tratamiento de textos, se sentirá motivado para prepararlos con mayor detenimiento. Por ejemplo, puede utilizar la orden INSERT para agregar una palabra, una oración entera o un párrafo, con la misma facilidad que si se tratara de una sola letra. Esto estimula al usuario para reconsiderar lo que está redactando en la carta o documento. La instrucción para eliminar texto es igualmente sencilla. Una orden hace que las palabras y letras que se desea suprimir desaparezcan de la pantalla y el texto restante se cierre, devolviendo a la página su aspecto impecable. Muchos escritores y periodistas profesionales en la actualidad utilizan procesadores de textos y, en general, opinan que su trabajo ha mejorado cualitativa y cuantitativamente.

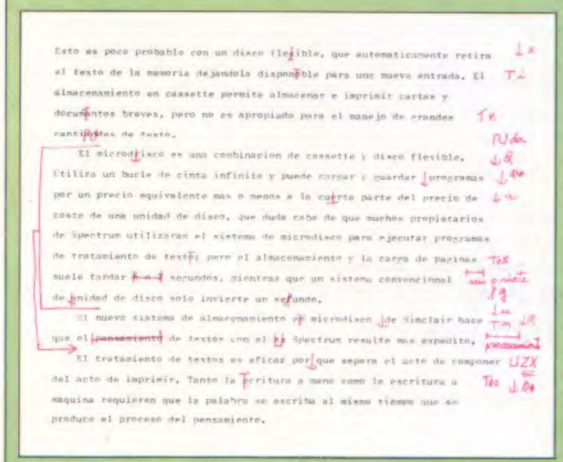


Estudiando el menú
La fotografía muestra el "menú" de un sofisticado paquete de programas para tratamiento de textos. El menú aparece en la pantalla apenas usted inserta el software, y le ofrece una guía sobre las diversas funciones de edición disponibles en el tratamiento de textos. Son ejemplos de funciones de edición: tabulación y determinación de márgenes, separación de las líneas, enumeración de palabras que integran el texto, reacomodación de párrafos y confección de un índice

Hasta los microordenadores más pequeños pueden ofrecer cierto potencial para tratamiento de textos. El Sinclair ZX81 puede operar con un programa sencillo para edición de textos, que le permite al usuario escribir una carta o un documento en pantalla y luego corregirlo. La frase "edición de textos" se aplica generalmente a un programa limitado para tratamiento de textos, que puede trabajar con una o dos páginas pero que es incapaz de tratar y almacenar documentos más extensos. La pequeña RAM del ZX81 y de los ordenadores de dimensiones similares limita drásticamente la cantidad de texto que puede visualizarse y con la cual es posible trabajar.

Una de las desventajas del ZX81 es que posee un teclado muy sensible al tacto, que impide una digitación rápida. Aunque el teclado del ZX Spectrum es bastante mejor, no llega a ser de la clase a la que están habituados los mecanógrafos. Si usted está estudiando la posibilidad de adquirir un micro para utilizarlo como procesador de textos, sería muy interesante que examinara los tipos de teclado, porque éstos tienen decisiva influencia en su comodidad de uso y en la velocidad de las pulsaciones (véase p. 32).

No obstante, un ordenador de 16 Kbytes o de 32 Kbytes puede resultar bastante útil para tratamiento

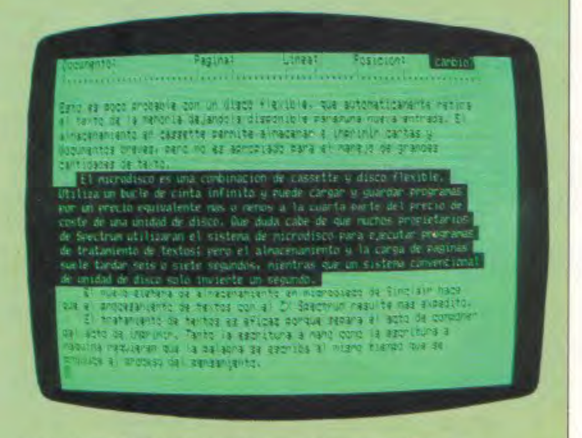


Esto es poco probable con un disco flesible, que automaticamente retira el texto de la memoria dejandola disponoble para una nueva entrada. El almacenamiento en cassette permite almacenar e imprimir cartas y documentos breves, pero no es apropiado para el manejo de grandes cantidades de texto.

El microdisco es una combinacion de cassette y disco flexible. Utiliza un bucle de cinta infinito y puede cargar y guardar programas por un precio equivalente mas o menos a la cuerta parte del precio de coste de una unidad de disco. Que duda cabe de que muchos propietarios de Spectrum utilizaran el sistema de microdisco para ejecutar programas de tratamiento de texto; pero el almacenamiento y la carga de paginas suele tardar 6 o 7 segundos, mientras que un sistema convencional de unidad de disco solo invierte un segundo.

El nuevo sistema de almacenamiento de microdisco de Sinclair hace que el pensamiento de textos con el Spectrum resulte mas expedito.

El tratamiento de textos es eficaz por que separa el acto de componer del acto de imprimir. Tanto la escritura a mano como la escritura a maquina requieren que la palabra se escriba al mismo tiempo que se produce el proceso del pensamiento.



Edición electrónica

La principal ventaja de un sistema de tratamiento de textos por ordenador respecto a la máquina de escribir, es la asombrosa adaptabilidad y velocidad que aquél ofrece. Mientras que los textos mecanografiados implican una laboriosa edición y corrección, mediante el tratamiento de textos puede hacerse lo mismo pero a velocidad electrónica. Y todas estas operaciones se realizan mientras el texto es visualizado en pantalla. Un sistema sofisticado puede buscar en el texto, cambiar palabras, trasladar líneas de un sitio a otro, revisar el texto automáticamente e, incluso, corregir su redacción y su ortografía. De hecho, un procesador de textos puede ahorrar tanto tiempo que gradualmente está comenzando a reemplazar a las máquinas de escribir convencionales. Compare las desordenadas correcciones del texto mecanografiado con la pulcra edición electrónica del procesador de textos

de textos. Después de comprar una impresora, el problema siguiente, en cuanto a hardware, consiste en disponer de suficiente memoria para almacenar sus textos. Un programa para tratamiento de textos se puede ejecutar utilizando una cassette para el almacenamiento. Sin embargo, el sistema de almacenamiento en cassette limita la cantidad de texto que es posible escribir porque la memoria se colma rápidamente. Esto es poco probable con un disco flexible, que automáticamente retira el texto de la memoria dejándola disponible para una nueva entrada. El almacenamiento en cassette permite almacenar e imprimir cartas y documentos breves, pero no es apropiado para el manejo de grandes cantidades de texto.

El nuevo sistema de almacenamiento en microdisco de Sinclair hace que el procesamiento de textos con el ZX Spectrum resulte más expedito. El microdisco es una combinación de cassette y disco flexible. Utiliza un bucle de cinta infinito y puede cargar y guardar programas por un precio equivalente más o menos a la cuarta parte del coste de una unidad de disco. Qué duda cabe de que muchos propietarios de Spectrum utilizarán el sistema de microdisco para ejecutar programas de tratamiento de textos; pero el almacenamiento y la carga de páginas suele tardar seis o siete segundos, mientras que un sistema convencional de unidad de disco sólo invierte un segundo.

El tratamiento de textos es eficaz porque separa el acto de componer del acto de imprimir. Tanto la escritura a mano como la escritura a máquina requieren que la palabra se escriba al mismo tiempo que se produce el proceso del pensamiento. En el caso del tratamiento de textos, en el papel no aparece ninguna palabra hasta que la composición en pantalla sea correcta. Pero llevar las palabras al papel exige una impresora, y el tipo de impresora barata apta para el listado de programas para ordenador es poco probable que pueda realizar una copia satisfactoria de una carta o un poema. (Véase la descripción de los tipos de impresora en la p. 74.)

Existe una gran competencia entre los fabricantes por producir sistemas baratos de tratamiento de textos, y ahora comienzan a ofrecerse algunos programas bastante económicos para ordenadores personales. Éstos almacenan las instrucciones de los programas para tratamiento de textos en un chip que puede enchufarse en la placa de circuito impreso del ordenador. Son muy útiles cuando no se dispone de una unidad de disco. Ofrecen la ventaja de que el programa puede cargarse rápidamente y utilizarse al instante, eliminando la espera que implica tener que cargarlo a través de un disco o de una cinta de cassette. Si la RAM del ordenador es suficientemente grande (a partir de 32 Kbytes), usted podrá escribir un texto de hasta 5 000 palabras y editarlo de la manera que le resulte más conveniente.

Si desea almacenar el texto editado después de haberlo impreso, necesitará guardarlo en una cinta de cassette, proceso que llevará algunos minutos. El software de tratamiento de textos barato no puede almacenar el texto que usted haya redactado. Si desea escribir una novela mediante un programa de tratamiento de textos, necesita conocer la capacidad del programa para manejar grandes cantidades de palabras.

Algunos sofisticados programas de tratamiento de textos pueden realizar funciones extras muy útiles. Una de las más populares es el diccionario automático o verificador de ortografía. Para utilizarlo necesitará un sistema de disco. El diccionario compara las palabras del texto con las que él tiene almacenadas. Señala las palabras que no reconoce y sugiere corregirlas.

La gran difusión del tratamiento de textos hace suponer que en el futuro será un ingenio esencial en la oficina y un dispositivo ideal para la correspondencia.

Microordenador	Software	En cassette	En disco	Teclado máq. escribir
Sinclair Spectrum	Wordprocessor	●	—	—
Commodore Vic 20	Vicwriter	●	—	●
Dragon 32	Gemini	●	—	●
New Brain	Propen	●	—	—
Commodore 64	Visawrite	—	●	●
Atari 800	Keyword	—	●	●
BBC Micro	View	—	●	●
Epson HX20	Deckmaster	●	—	●

Elección del procesador de textos

Si usted piensa utilizar su procesador de textos durante períodos muy largos, asegúrese de que podrá digitar cómodamente en el teclado de su ordenador. Los teclados de tipo membrana, similares a los de las máquinas de calcular, se desarrollaron para reducir los costos de fabricación y son más adecuados para juegos y para escribir programas cortos. También es una ventaja que su ordenador cuente con teclas especiales de función programables. Éstas se emplean con frecuencia en los paquetes de programas para tratamiento de textos más sofisticados, y reducen la cantidad de órdenes que usted debe digitar en su teclado. Obviamente, si desea copias impresas es indispensable comprar una impresora. Las impresoras varían notablemente en cuanto a calidad de impresión y velocidad de ejecución (véase p. 74). Asegúrese de que su procesador de textos y su impresora sean compatibles para las diversas tareas que les asigne. Si lo que usted desea es escribir cartas, puede interesarle una impresora y un procesador de textos de precio razonable. Pero para textos más largos necesitaría una combinación más cara

Preguntas y respuestas

¿Qué es la "quinta generación de ordenadores" y cuáles fueron las cuatro anteriores?
La quinta generación es la etapa de desarrollo en el campo de los ordenadores en la cual trabajan actualmente los ingenieros y programadores. Se espera que estos ordenadores representen un salto significativo respecto a la tecnología actual.

El término *quinta generación* lo acuñaron los japoneses para describir sus proyectos de investigación a largo plazo. Los ordenadores de la quinta generación no tendrán teclados y no necesitarán ser programados en lenguajes como el BASIC y el PASCAL. En cambio, se espera que el hombre pueda hablarle al ordenador y que éste le responda en cualquier idioma. También es probable que estos ordenadores de quinta generación escriban sus propios programas como solución a los problemas que les sean planteados.

Los ordenadores de la primera generación fueron los primeros dispositivos para cálculo enteramente electrónicos, y se inventaron poco después de finalizada la segunda guerra mundial. Los de la segunda generación se basaban esencialmente en los mismos conceptos, pero en vez de válvulas y relés tenían transistores, por lo cual eran más pequeños y más baratos. Los de la tercera generación utilizaban circuitos integrados (circuitos electrónicos montados sobre silicio). Estos circuitos fueron el primer antecedente del microchip y marcaron el comienzo del descenso de los precios de los ordenadores. No obstante, aún resultaban muy caros para un hogar o para una oficina corrientes.

La cuarta generación corresponde a la tecnología actual. Estos ordenadores utilizan circuitos LSI (Large Scale Integration, integración a gran escala). El desarrollo de estos microchips ha hecho posible que el ordenador esté al alcance de la mayoría de las personas.



¿Dónde está Silicon Valley?
Se conoce como Silicon Valley (Valle del Silicio) a una zona que se extiende desde San José hasta el sur de San Francisco (California), donde la mayoría de las empresas norteamericanas dedicadas a la microelectrónica y a la fabricación de ordenadores tienen sus sedes o sus centros de investigación. No existe razón alguna para que en esta área tan pequeña se haya concentrado tanta experiencia humana. ¡No hay ninguna fuente de recursos naturales que favorezca la fabricación de microchips! Hasta hace 20 años la zona sólo se conocía por su producción frutícola.



¿Es necesario todo ese "lío" de cables sueltos que tienen por detrás algunos ordenadores?
La mayoría de los microordenadores actuales se diseñan atendiendo tanto al aspecto exterior como al aspecto puramente electrónico. Ese "lío" de cables por lo general se esconde. Pero, en el caso de algunos ordenadores para investigación muy avanzados, estos cables sueltos son muy importantes. La electricidad se desplaza a la velocidad de la luz; no obstante, necesita cierto tiempo para recorrer un cable. Estos ordenadores trabajan con tanta rapidez que la información debe llegar al lugar adecuado en el momento preciso. La longitud de los cables se calcula con exactitud para asegurar una sincronización perfecta.



En la publicidad de los ordenadores suele indicarse que éstos poseen un microprocesador Z80 o 6502. ¿Qué significan estas cifras?
Las cifras en sí mismas no significan nada; "6502" es sólo la referencia de identificación o el "nombre" de un determinado chip de microprocesador, y "Z80" corresponde a otra referencia. Todos los ordenadores basados en el mismo microprocesador comprenden idéntica serie de instrucciones básicas (denominada *código de lenguaje máquina*), a partir de las cuales se construyen los programas. Sin embargo, el usuario por lo general escribe sus programas en un lenguaje de alto nivel, como el BASIC, que luego es interpretado por el ordenador en código de lenguaje máquina. Por lo tanto, a menos que usted desee escribir programas directamente en código de lenguaje máquina, no tiene ninguna importancia qué clase de microprocesador posea su ordenador.

A pesar de que algunos tipos de microprocesadores funcionan a mayor velocidad que otros, la velocidad a la que se desarrolle un programa depende muchísimo más de la forma en que haya sido escrito el software.


David Higham

¿Cómo pueden los ordenadores ayudar en la lucha contra la delincuencia?
Para formarnos una idea de la labor que pueden desempeñar los ordenadores en este campo, informaremos someramente acerca del uso que la policía de Gran Bretaña está realizando de los avances de la informática. En 1968 se organizó una unidad de informática (Police National Computer Unit) y en 1974 se instaló el primer archivo de información (de vehículos robados y sospechosos). Hoy en día son pocos los coches de la policía que poseen terminales de ordenador a través de los cuales se pueda acceder a la información del ordenador central. Dado el nivel actual de la tecnología de almacenamiento, sería posible que la policía británica almacenara los detalles personales de todos los ciudadanos. Por ahora, el gobierno británico ha organizado diversas comisiones que actúan como "guardianes" de los intereses de las personas.

Micros en marcha

Basta de consultar mapas de ruta y de preocuparse del indicador de la gasolina: el coche del futuro lo llevará a su punto de destino de forma económica y segura



Un dinámico tablero de instrumentos
Puede que el tablero de instrumentos de su próximo coche se parezca al que muestra la fotografía. Todos los diales y los indicadores móviles utilizados en los tableros de instrumentos convencionales han sido reemplazados por un terminal especialmente diseñado, controlado por ordenador. Los diodos luminosos pueden controlar la velocidad, el nivel de gasolina y la temperatura con mayor velocidad y exactitud que los instrumentos electromecánicos

Cortesía de Toyota

La fabricación y el diseño de coches será uno de los campos donde veremos funcionando claramente la tecnología de los ordenadores. En la actualidad ya es posible adquirir un automóvil que detecte él mismo el momento en que necesite un servicio de mantenimiento y que avise cuándo es preciso visitar el taller.

El coche puede poseer esta facultad porque unos sensores acoplados en diversos puntos del motor alimentan a un microprocesador con información acerca del kilometraje y las temperaturas, permitiendo que evalúe las condiciones en que está siendo utilizado el vehículo. El conductor sabrá que hace falta una reparación porque el diminuto ordenador hace funcionar una serie de luces verdes, ámbar y rojas, situadas en el tablero de instrumentos. Cuando se apagan las luces verdes y se encienden las rojas, el conductor es advertido de que es necesario llevar el coche a un taller.

En Europa y en Japón se fabrican coches que le hablan al conductor, indicándole que se coloque el cinturón de seguridad o avisándole de la inminencia de un problema en el motor, como un recalentamiento o la disminución de los niveles de agua y aceite. El coche puede hablar porque posee un sintetizador de voz; se trata de un ordenador que ha sido programado digitalmente con las características de la voz humana. Esto se conoce como *sonido digital*, porque las ondas sonoras se transforman en números binarios, para que resulten comprensibles al ordenador.

Cuando los sensores del motor detectan una situación de alarma, el ordenador activa el sintetizador de voz que, a su vez, convierte la salida digital en el sonido de una voz humana y la emite por un altavoz.

Un viaje tranquilo

Otra de las formas en que los ordenadores pueden contribuir a mejorar los coches es a través del control de la suspensión. La Lotus está trabajando en un sistema denominado "suspensión activa". Esta técnica utiliza un ordenador para regular la rigidez y la flexibilidad de los amortiguadores de coche en fracciones de segundo; de esta manera el vehículo conserva su estabilidad en la carretera tanto si el conductor viaja solo como si lo hace con pasajeros y equipaje.

Los coches deportivos tienen, por lo general, una suspensión rígida que les permite afirmarse bien sobre la carretera. El inconveniente de este tipo de suspensión es que la marcha es muy dura y los pasajeros sienten todos los baches. Pero si un coche tiene una suspensión blanda que le proporcione una marcha "entre algodones", no tomará bien las curvas, porque la carrocería se balanceará demasiado. Con el control de la suspensión por ordenador se logrará aunar las ventajas de ambos sistemas.

Las firmas japonesas Honda y Toyota también están desarrollando ordenadores para navegación. Estos ordenadores le señalan al piloto el rumbo a tomar. Y lo hacen midiendo la velocidad, la dirección y la distancia recorrida, al tiempo que comparan estos datos con un mapa de la ruta correcta que llevan en su memoria. El conductor puede entonces decidir si gira hacia la izquierda o hacia la derecha o si sigue adelante, de acuerdo a una serie de indicadores en el tablero de instrumentos.

Uno de los ordenadores más comunes en los coches mide el consumo de gasolina y puede calcular la hora de llegada. El conductor sabe, en un momento dado, cuánta gasolina está consumiendo, mientras el ordenador le indica cuál ha sido la velocidad promedio del viaje. Uno de los ordenadores para automóviles más avanzados permite que el conductor programe una velocidad de crucero. El coche, entonces, mantendrá esta velocidad sin que para ello el conductor necesite tocar el acelerador.

Los ordenadores que se instalan en *juggernauts* de larga distancia tienen una finalidad muy diferente. Sirven como cuaderno de bitácora electrónico y permiten que las autoridades de tráfico puedan determinar durante cuánto tiempo ha permanecido al volante el conductor y a qué velocidad, así como la distancia recorrida.

Una de las ventajas más importantes que ofrecen los ordenadores para coches es el obtener un mejor aprovechamiento del combustible. La firma alemana BMW ya posee una gama de automóviles con un sistema que determina la mezcla óptima de gasolina y aire requerida en un momento dado de la conducción.

En efecto, se trata de "sintonizar" al coche muchísimas veces por segundo para racionalizar todo lo posible el consumo de combustible. El sistema funciona midiendo continuamente la mezcla de aire y gasolina y realizando ajustes para tener presente la velocidad del coche, en qué marcha está trabajando y la temperatura del motor.

El futuro
¿Qué nos depara el futuro respecto a los ordenadores para automóvil? Teóricamente, sería posible que el ordenador se hiciera cargo por completo del acto de la conducción en sí mismo. Todo cuanto el conductor debería hacer sería programar el ordenador del coche con el punto de destino. El ordenador guiaría automáticamente el coche valiéndose de la información proveniente de sensores instalados en la carretera o comunicándose con los ordenadores centrales de tráfico. Otro desarrollo que podremos ver es el radar por ordenador, que reajustaría automáticamente la velocidad del vehículo si éste se acercara demasiado al coche que lo precediera.
Pronto los instrumentos de dial serán reemplazados por un tablero de instrumentos con una pantalla similar a la de un monitor en la cual se visualizarán los gráficos trazados por el ordenador. El conductor podrá solicitar la visualización electrónica que desee, como la temperatura del motor o el nivel de gasolina. La información relativa a la conducción podrá proyectarse sobre el parabrisas, de modo que el usuario no necesitará apartar sus ojos de la carretera. También es posible que los coches vengan equipados con ordenadores que informen de inmediato al mecánico acerca de cualquier tipo de problema. Se podría conectar el ordenador del coche al ordenador preprogramado del taller, que realizaría una revisión general. La fotografía muestra el prototipo de la Honda de un sistema de navegación por ordenador, montado en el tablero de instrumentos



Ian McKinnell

Mensaje comprendido

Si se pulsa una tecla, de inmediato entran en acción estratos ocultos de software, que decodifican las instrucciones, buscan en la memoria y exploran el teclado a la espera de la próxima orden

Un ordenador es un ensamblaje de metal, plástico y silicio que, si carece de algún programa en su memoria, es incapaz de realizar ninguna clase de tarea útil; algo así como un tocadiscos sin ningún disco en el plato. El proceso para lograr que el ordenador realice la tarea específica que usted necesita se conoce como "programación". Incluso un recién iniciado en programación podrá identificar dos fases diferentes para resolver un problema. En primer lugar, éste debe traducirse y escribirse de forma tal que el ordenador pueda comprenderlo. En segundo lugar, este programa debe alimentarse al ordenador y "ejecutarse". En un segundo momento, estas dos fases se subdividen en dos etapas: la primera le corresponde al propio programador, mientras que en la segunda etapa es el ordenador el que debe emprender las acciones (por lo general sin el conocimiento o la intervención del usuario).

Supongamos que desea escribir un programa para preparar una nómina. Lo primero que necesita es comprender perfectamente el problema. ¿Qué salida requiere usted del ordenador? ¿Qué información necesitará el ordenador para realizar los recibos de pago semanales? Esto puede suponer información acerca de salarios, horas trabajadas por semana, etc. El segundo elemento esencial consiste en especificar el proceso por el cual se ha de producir esta salida, por ejemplo: "¿Cómo se calculan las retenciones para impuestos y los descuentos para la Seguridad Social?"

Si se trata de una aplicación para una gran empresa, esta tarea la puede efectuar un experimentado "analista de sistemas", cuya especialidad es la de analizar la forma en que funciona una empresa y en escribirla de manera que pueda traducirse fácilmente en un programa. En el caso de programas domésticos o educativos, esta labor podrá realizarla el propio programador.

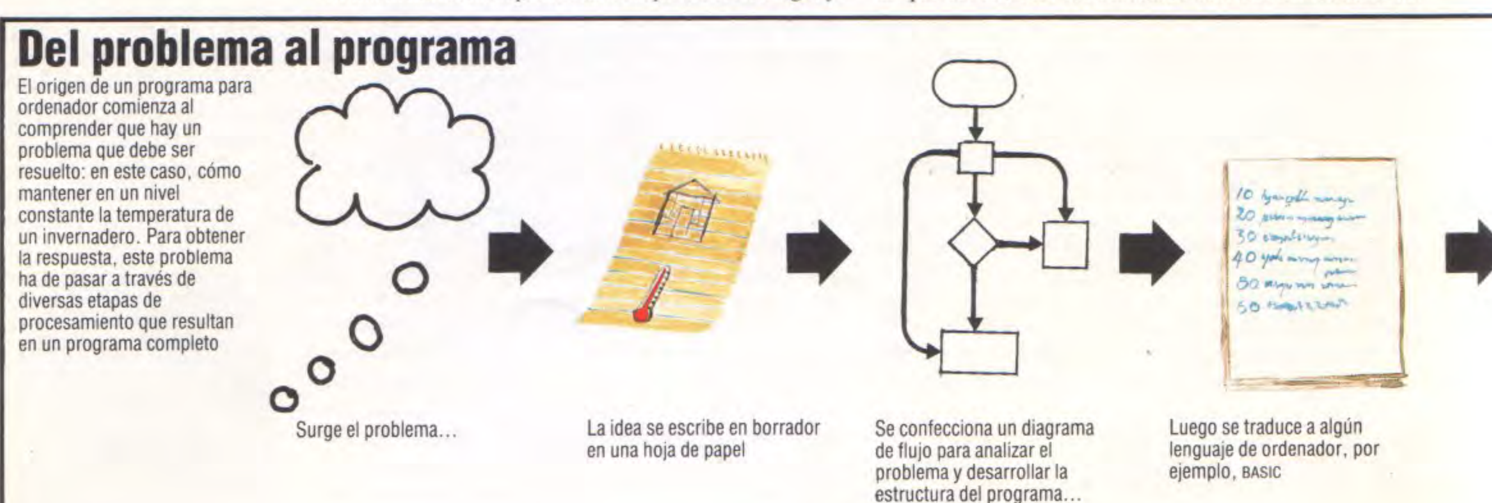
Si los ordenadores pudieran comprender el lenguaje corriente, todas estas "especificaciones de programa" podrían ejecutarse directamente; pero, por desgracia, los ordenadores todavía no lo comprenden. Muchos principiantes encuentran dificultades porque tratan de escribir el programa desde el comienzo hasta el final, del mismo modo en que se traduciría al francés un ensayo escrito en castellano. No obstante, algunos programadores muy experimentados dividen aún más esta etapa. Éstos podrían fraccionar la especificación de la nómina en cuatro "módulos": para la entrada de los datos de la semana, para cálculo, para el almacenamiento de los resultados acumulables (como "impuestos pagados este año") y para imprimir las hojas de salarios.

Cada módulo puede, entonces, dividirse en estructuras más pequeñas. Esto se conoce como "programación estructurada"; cada una de estas secciones más pequeñas es sencilla y puede expresarse en una o dos líneas del programa. Por último, todo el grupo de líneas (el listado del programa) se digita en el ordenador.

Un buen programador siempre conserva notas de cada etapa y éstas reflejan los muchos niveles que distinguen a un programa escrito en lenguaje corriente de un programa escrito en un lenguaje de alto nivel como el BASIC.

Lo que sucede a partir del momento en que se pulsa RUN queda completamente bajo el control del ordenador y, otra vez, vuelve a suponer muchas y diferentes etapas o estratos. Sin embargo, las operaciones internas del ordenador están "ocultas": lo único que el usuario percibe es que su programa le está solicitando alguna información complementaria y está produciendo la salida requerida.

Debido a que el microprocesador no puede comprender un lenguaje de alto nivel, la primera tarea con la que se enfrenta el ordenador consiste en traducir las



Del problema al programa

El origen de un programa para ordenador comienza al comprender que hay un problema que debe ser resuelto: en este caso, cómo mantener en un nivel constante la temperatura de un invernadero. Para obtener la respuesta, este problema ha de pasar a través de diversas etapas de procesamiento que resultan en un programa completo

Surge el problema...

La idea se escribe en borrador en una hoja de papel

Se confecciona un diagrama de flujo para analizar el problema y desarrollar la estructura del programa...

Luego se traduce a algún lenguaje de ordenador, por ejemplo, BASIC

instrucciones en código de lenguaje máquina. En los ordenadores personales esta tarea la realiza el interpretador que está almacenado permanentemente en la ROM de la máquina.

El interpretador es un sofisticado programa en código de lenguaje máquina que el microprocesador ejecuta directamente. Cuando se digita RUN, el interpretador comienza a examinar el programa del usuario, carácter por carácter. Compara todas las frases que encuentra con las de su propio diccionario. Si se encuentra con un carácter que no comprende (lo que puede suceder simplemente porque usted ha cometido un error de digitación), dejará de tratar de interpretar el programa e imprimirá en pantalla un mensaje indicando SYNTAX ERROR (error de sintaxis).

Si la palabra está incluida en el diccionario del interpretador (por ejemplo, PRINT), pasa inmediatamente a la parte del interpretador que sabe cómo tratar esta función. En este caso, la rutina examinará luego lo que venga a continuación de la palabra PRINT en el programa del usuario y preparará esta información para ser visualizada en forma de flujo de caracteres.

En este punto comienza a operar el siguiente nivel. En algún otro lugar de la memoria del ordenador hay una rutina que puede aceptar un flujo de caracteres, almacenarlos en otra zona de la memoria reservada para la pantalla y ordenarlos para que puedan ser convertidos en la clase de señales que necesita la pantalla de televisión o el monitor. Esto es algo que debe hacerse continuamente, incluso cuando el programa en sí mismo está comprometido sólo en cálculos.

Lo mismo puede decirse respecto al otro extremo del ordenador: el teclado. En el interior del ordenador, una rutina de programa escrita especialmente debe explorar el teclado para averiguar si han sido pulsadas algunas teclas y, en caso afirmativo, colocar los códigos adecuados en otra zona de la memoria para su utilización como la entrada del programa del usuario. Y, puesto que usted puede desear interrumpir en cualquier momento la realización del programa pulsando la tecla BREAK, el teclado debe ser explorado continuamente, aun mientras el programa se está ejecutando.

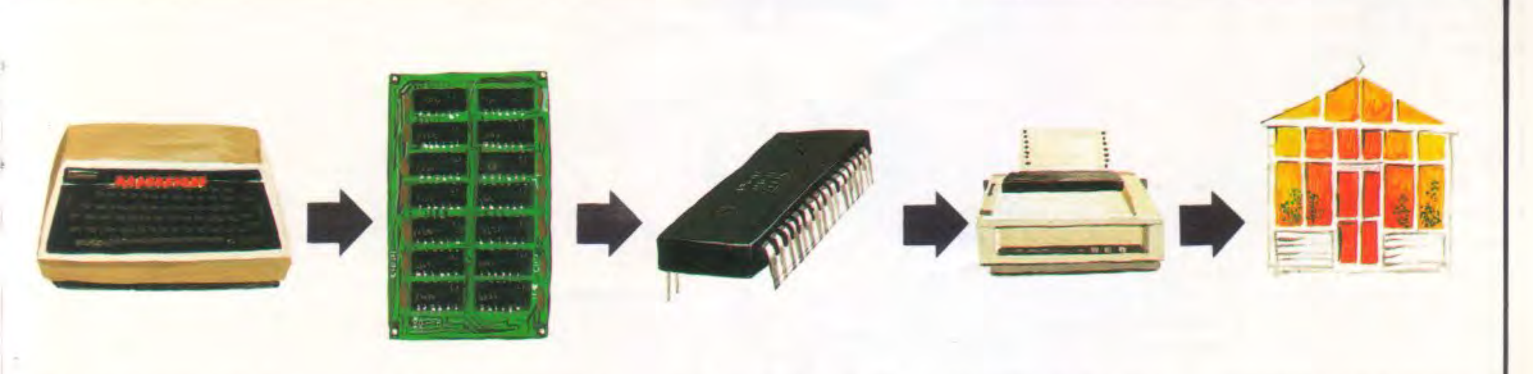
En realidad, el microprocesador que poseen la mayoría de los ordenadores personales no puede llevar a cabo más de una tarea a la vez, de modo que, efectivamente, ha de dividir su tiempo entre interpretar el programa del usuario y realizar sus propias funciones internas, tales como verificar el teclado y controlar la pantalla. Esta tarea la puede cumplir mediante un procedimiento "por interrupción"; un circuito electrónico especial interrumpe al microprocesador unas 50 veces por segundo y "le recuerda" que debe realizar sus tareas propias y otras funciones en la pantalla y el teclado, antes de continuar con lo que estaba haciendo.

Así, pues, incluso cuando su programa haya sido digitado, el ordenador ha de llevar a cabo muchos niveles de procesamiento antes de producir los resultados. Si bien el proceso puede resultar complicado, es el ordenador quien se ocupa de la mayor parte del mismo.

La tendencia actual se orienta a facilitar la tarea del programador, haciendo recaer en el ordenador la realización de la mayor cantidad posible del trabajo de rutina. Los ordenadores de la próxima generación podrán escribir ellos mismos programas completos a partir de unas especificaciones en lenguaje corriente.



El software oculto
En todo ordenador existe una compleja jerarquía de software oculto que trabaja constantemente. Entre sus muchas tareas, controla y verifica cuándo ha sido pulsada una tecla y de cuál se trata, qué se está visualizando en pantalla, qué instrucciones se les están proporcionando a los periféricos, y el estado y contenido de la memoria RAM. Todas estas funciones se desempeñan de manera incesante mientras el usuario está concentrado en la siguiente función de su programa. El mismo principio de la jerarquía de software oculto rige tanto para los sofisticados ordenadores de gestión (como el de la fotografía) como para los ordenadores personales



El programa se introduce en la memoria del ordenador a través del teclado

El programa en BASIC pasa a través de una matriz de chips. Éstos lo convierten en código de lenguaje máquina y lo retransmiten a la CPU

El cálculo se realiza en la CPU. La información resultante se transmite posteriormente a un periférico (p. ej., una impresora, un monitor, etc.)

En este caso la impresora produce una salida impresa o "copia impresa"

Si el programa es exacto, el problema ya está resuelto

¿Verdadero o falso?

Quizá los ordenadores aún no sepan "pensar"; pero saben seguir las leyes de la lógica

Acerca de la CPU (unidad central de proceso) suele decirse que es el corazón del ordenador. Es el lugar donde se efectúan todos los cálculos y donde se toman todas las decisiones lógicas. Pero, ¿cómo se efectúan estos cálculos y cómo se adoptan las decisiones?

Para llegar a comprenderlo, es preciso conocer los principios básicos de la aritmética binaria y familiarizarse con las puertas lógicas. En los ordenadores, estas puertas son circuitos eléctricos simples que pueden adoptar decisiones lógicas y establecer comparaciones lógicas. Esto parece más complicado de lo que en realidad es, y los principios pueden ilustrarse a partir de ejemplos tomados de la vida cotidiana.

Existen tres tipos de puertas fundamentales: la puerta AND (y), la puerta OR (o) y la puerta NOT (no), que se escriben con letras mayúsculas para diferenciarlas de las palabras normales en inglés "y", "o" y "no".

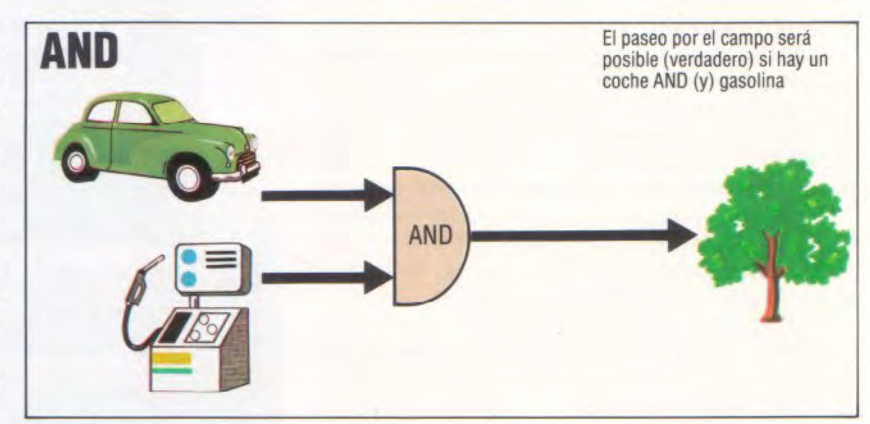
Conexiones lógicas

Una puerta AND es un circuito que produce una salida "verdadera" si todas las entradas son "verdaderas". Veamos qué significa esto. Supongamos que quisiera dar un paseo al campo. Si tiene un coche AND (y) gasolina, podrá dar ese paseo. No podría darlo si tuviera gasolina pero no poseyera un coche. Igualmente, tampoco sería factible dicho paseo si tuviera coche pero no gasolina.

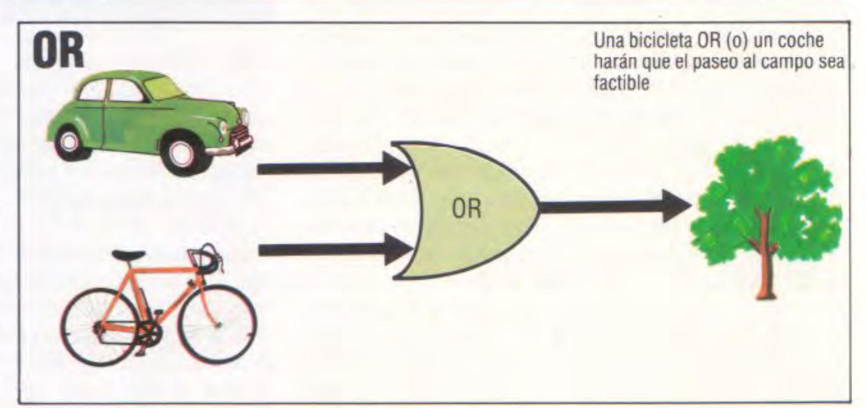
En este "circuito" AND hay dos condiciones de entrada y ambas han de ser "verdaderas". Para que pueda dar el paseo (la "salida") ha de ser verdad que posee un coche AND (y) ha de ser verdad que tiene gasolina. Entonces la salida se convierte en "verdadera": es verdad que puede dar un paseo por el campo. Más adelante veremos cómo este diagrama lógico puede expresarse como una ecuación lógica, y cómo puede representarse en una "tabla de verdad".

Imaginemos una situación ligeramente diferente. A alguien le agradaría dar un paseo por el campo. El paseo será posible si esa persona posee un coche OR (o) una bicicleta (esta vez damos por sentado que el coche tiene gasolina). Si posee un coche, podrá ir de paseo. Si tiene una bicicleta, podrá ir. Sólo en el caso de que ninguna de las condiciones de entrada fuese verdadera sería imposible el paseo; en el lenguaje de informática, la salida se convierte en falsa (es decir, no es verdad que esta persona pueda ir de paseo al campo).

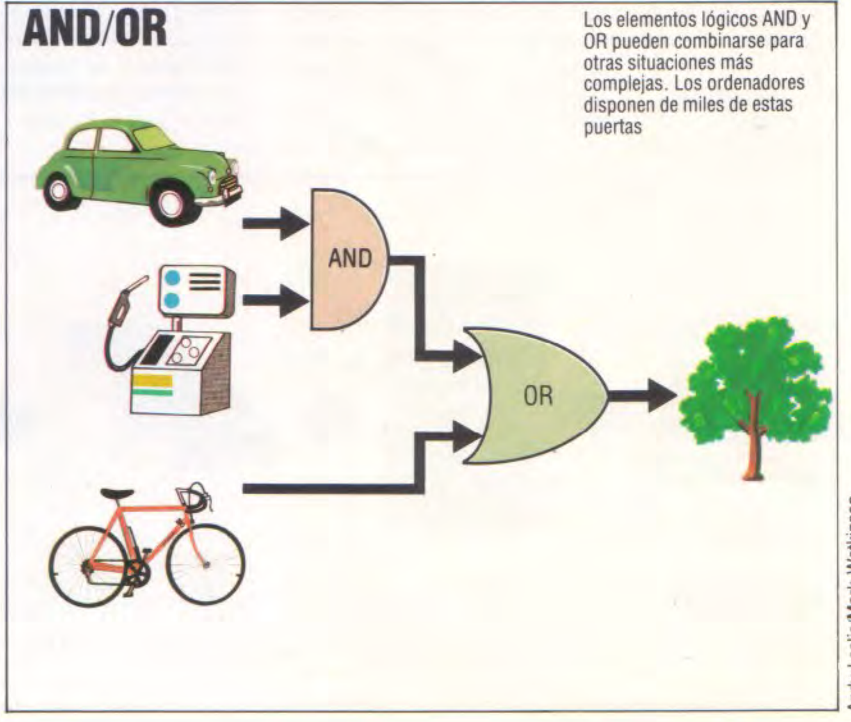
Nos queda por considerar otra puerta lógica esencial: la puerta NOT (no). Esta puerta simplemente da como salida lo contrario de la entrada. Si la entrada es verdadera, la salida será falsa. Si la entrada es falsa, la salida será verdadera. Extendiendo nuestra metáfora del paseo al campo, ya sea en coche o en bicicleta, debe ser falso que tenemos un neumático desinflado para dar dicho paseo. Si la entrada (neumático desinflado) fuera verdadera, la salida (paseo) sería falsa.



El paseo por el campo será posible (verdadero) si hay un coche AND (y) gasolina

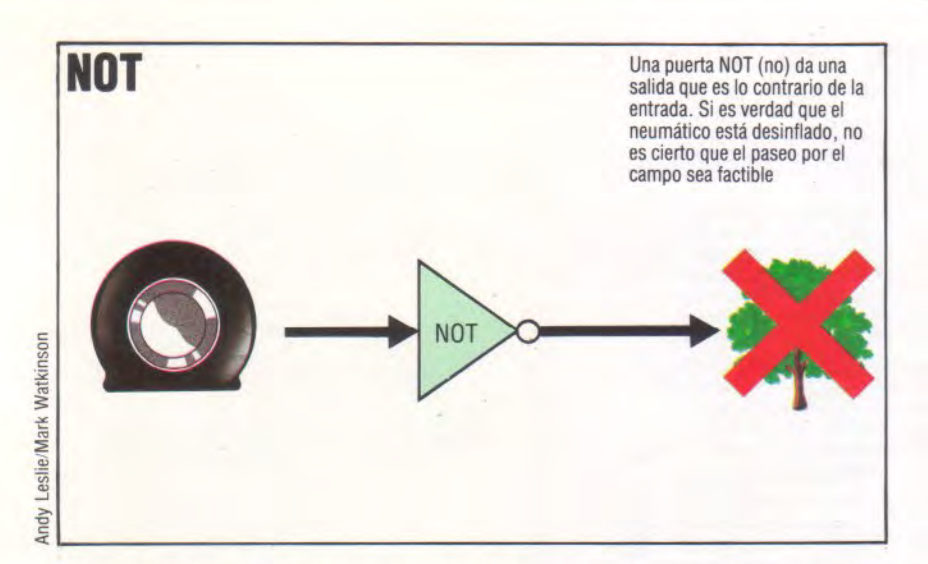


Una bicicleta OR (o) un coche harán que el paseo al campo sea factible



Los elementos lógicos AND y OR pueden combinarse para otras situaciones más complejas. Los ordenadores disponen de miles de estas puertas

Andy Leslie/Mark Watkinson



Una puerta NOT (no) da una salida que es lo contrario de la entrada. Si es verdad que el neumático está desinflado, no es cierto que el paseo por el campo sea factible

Andy Leslie/Mark Watkinson

Estos elementos lógicos pueden combinarse entre sí y lo hemos demostrado valiéndonos del ejemplo del paseo por el campo. Las combinaciones entre AND (y), OR (o) y NOT (no) permiten tomar todas las decisiones sobre la base de la lógica convencional. Es interesante elaborar las decisiones lógicas (puertas) que se necesitarían para otros problemas. Trate de elaborar, por ejemplo, lo que se requeriría para hacer una barbacoa en el jardín. Puede resultar bastante complicado. Para tener una barbacoa en el jardín (la salida verdadera) necesitaríamos varias condiciones de entrada: dinero OR (o) un talonario de cheques OR (o) una tarjeta de crédito (para comprar la comida y la bebida) AND (y) una tarde libre AND (y) buen tiempo AND (y) una parrilla AND (y) carbón.

Las tablas de verdad

Los símbolos que hemos usado en las ilustraciones son los mismos que se utilizan en los diagramas de los circuitos de ordenador. Para ver con qué facilidad se pueden realizar decisiones lógicas utilizando circuitos eléctricos, observemos la "tabla de verdad" para la ilustración AND. Si empleamos la letra *c* para representar la condición de entrada "tener un coche" y la letra *g* para la condición de entrada "tener gasolina", podemos representar la condición de salida "dar un paseo por el campo" empleando la letra *p*. Luego podremos utilizar la letra V para representar "verdadero" y la letra F para representar "falso". La tabla de verdad muestra todas las combinaciones posibles de condiciones de entrada y los efectos de la utilización de AND en la salida. Sería así:

COCHE	F	V	F	V	(c)
GASOLINA	F	F	V	V	(g)
PASEO	F	F	F	V	(p)

Tabla de verdad para dos entradas AND (y)

COCHE	0	1	0	1	(c)
GASOLINA	0	0	1	1	(g)
PASEO	0	0	0	1	(p)

La misma tabla de verdad utilizando 0 y 1 para "falso" y "verdadero"

En los ordenadores utilizamos los dígitos binarios cero y uno para representar, respectivamente, falso y verdadero. El ordenador interpreta un voltaje positivo como un uno y un voltaje negativo como un cero. Un circuito AND puede montarse fácilmente utilizando transistores, de manera que si ambas entradas son voltajes positivos, la salida también será un voltaje positivo. Si una o ambas entradas son de un voltaje negativo, la salida del circuito será igualmente negativa.

Un circuito electrónico OR produce una salida de voltaje positiva si una o ambas entradas son positivas. Si ambas entradas son negativas, la salida también será negativa. En un circuito NOT, la entrada simplemente se invierte: si la entrada es positiva, la salida será negativa; si la entrada es negativa, la salida será positiva.



El chip 7408
Los grandes chips de silicio suelen contener miles de puertas para efectuar la lógica AND, OR y NOT. El pequeño chip 7408 que aquí vemos contiene todos los transistores y los sistemas de circuitos para cuatro puertas AND. Las puertas están representadas mediante los símbolos lógicos, en vez de mostrar los verdaderos circuitos

Tony Lodge

Dragon 32

Un ordenador galés que incorpora un chip nuevo y una serie de sofisticados gráficos

Dragon Data es una compañía galesa, respaldada por la Welsh Development Agency y el Prudential Group, pero fundada por los fabricantes de juguetes Mettoy.

El Dragon 32 salió al mercado en Gran Bretaña durante las Navidades de 1982, y obtuvo un éxito inmediato gracias a sus 32 Kbytes y a su intérprete de lenguaje BASIC.

El Dragon 32 es altamente compatible con el ordenador Tandy Color; se pueden utilizar los complementos y algunos de los cartuchos de juegos de la marca Tandy, pero no las cassettes. Las dos máquinas emplean el mismo microprocesador, el Motorola 6809E (véase recuadro), en vez del 6502 y el Z80, como la mayoría de los ordenadores personales.

Una serie de órdenes para los gráficos le dan un control mayor que el que proporcionan otras máquinas con una mejor resolución. Algunos ejemplos de estas órdenes son: DRAW, CIRCLE, PAINT, COLOUR y MOVE.

El Dragon sólo puede tocar una nota al mismo tiempo, a diferencia de otros ordenadores con más de una "voz" y que pueden generar acordes. Pero sus órdenes de BASIC le permiten crear una melodía identificable con mayor facilidad que la mayoría de las otras máquinas.

Aunque está bien equipado con interfaces, el número de dispositivos para ampliar el Dragon 32 es limitado, sin tener en cuenta las palancas de mando.

El proyecto de ampliar la capacidad de memoria a 64 Kbytes permitiría su utilización con pequeños programas de gestión, lo que lo haría entrar en competencia con ordenadores tales como el Commodore 64.

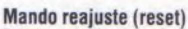


Unidad de disco Dragon
Esta unidad consta de dos partes: una tarjeta controladora de discos, que se introduce en la carga de cartuchos del Dragon, y el dispositivo principal, que incluye una unidad de disco

Chris Stevens

Teclado
El Dragon 32 tiene un teclado semejante al de una máquina de escribir.
Las teclas de movimiento del cursor hacia arriba/abajo y derecha/izquierda están inadecuadamente situadas en ambos extremos del teclado. La tecla BREAK interrumpe el funcionamiento de un programa y CLEAR limpia la pantalla. Aunque su diseño sugiere que en la parte superior de la cubierta se puede colocar un pequeño televisor, esto no es recomendable

Mando reajuste (reset)
Produce el mismo efecto que desconectar y volver a conectar el ordenador, pero hace que el consumo de energía sea menor



Conexión cassette
Se puede conectar una cassette normal. El BASIC del Dragon 32 permite el control del motor de la cassette

Conexión palanca de mando
El Dragon 32 puede funcionar con dos palancas para juegos

Conexión impresora
Una interface estándar de ocho bits en paralelo permite que el Dragon 32 funcione con la mayoría de las marcas de impresoras

Controlador video
Este chip genera, a partir de los caracteres y símbolos almacenados en la memoria, las señales de video necesarias para una pantalla de un televisor o un monitor

Modulador RF
Transforma las señales de video para que puedan ser captadas adecuadamente, a través de la toma de antena, por un aparato de televisión

Interface de video
Desde este punto se suministran, por separado, las señales en rojo, verde y azul para un monitor adecuado. El sistema produce resultados de la mejor calidad.

Interruptor principal

Chips interface
Dos adaptadores interface periféricos 6821 (PIA) realizan todas las conversiones de señal, desde la CPU hasta el teclado, cassette y dispositivos externos, por ejemplo una impresora



Cristales de cuarzo
Suministran las señales de tiempo de modo que todos los componentes del ordenador funcionen de forma sincronizada. Uno de ellos controla la velocidad del microprocesador; el otro, principalmente, los circuitos de video

RAM
La RAM estándar del Dragon 32, de 32 K, está formada por 16 chips de 2 K

ROM
Los dos chips ROM contienen el sofisticado intérprete Microsoft BASIC, el sistema operativo que rige todas las funciones internas del ordenador, por ejemplo la transferencia de información desde el teclado a la pantalla

Microprocesador
El Motorola 6809E es más rápido que el más popular 6502, aunque el software para aquél es menor. En el ordenador Tandy Color se emplea el mismo chip, lo cual explica que ambas máquinas sean altamente compatibles

Conexión principal expansión
Las patillas de esta interface se enchufan directamente al microprocesador; por ello se usa principalmente para conectar cartuchos de software

Reguladores de voltaje
Un transformador exterior reduce el voltaje al apropiado para ser utilizado en el interior del ordenador. Estos reguladores, con los capacitadores cilíndricos situados junto a ellos, son necesarios para eliminar cualquier variación o fluctuación del voltaje. Para evitar su sobrecalentamiento, están montados sobre un disipador de calor metálico

Enchufe red
En él se conecta el transformador principal

El chip 6809E
El Dragon 32 utiliza el microprocesador de 8 bits 6809E, en vez del más popular 6502 o Z80 usados por otros muchos ordenadores. El 6809E es altamente apreciado por muchos programadores. Es fácil escribir programas para el 6809E empleando su propio lenguaje máquina

DRAGON 32

DIMENSIONES
380 × 325 × 97 mm

PESO
2,1 kg

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
La RAM existente de 32 K será incrementada, posteriormente, a 64 K (de los cuales sólo 48 K se podrán utilizar para BASIC). Hay 16 K de ROM que contienen el intérprete de Microsoft BASIC y sistema operativo

VISUALIZACION EN VIDEO
Puede contener 16 líneas con 32 posiciones de caracteres. Se dispone de un total de ocho colores, que se reducen a dos para la máxima resolución de gráficos de 256 × 192 puntos

INTERFACES
Cassette, palancas de mando (dos conexiones), monitor de televisión, carga de cartuchos, interface paralela para impresora, etcétera

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
Ninguno

VIENE CON
Unidad suministradora de potencia, cable aéreo, manual de instrucciones

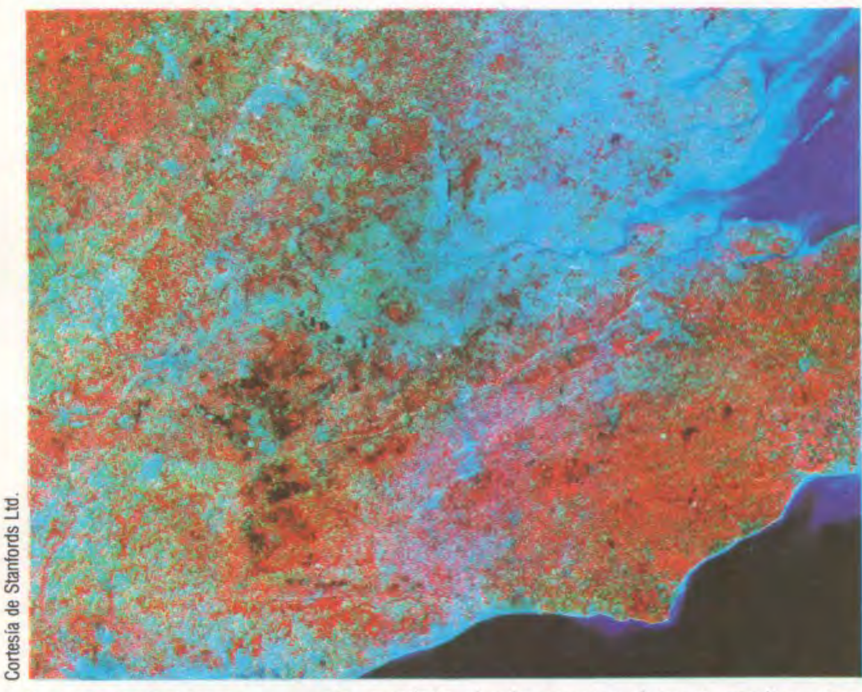
TECLADO
Disposición tipo máquina de escribir con 53 teclas móviles

DOCUMENTACION
El manual del usuario, escrito por el constructor del ordenador, Richard Wadman, ha sido realizado a un alto nivel.
Todas las respuestas del ordenador están impresas en diferente color, y contiene numerosas indicaciones y consejos útiles.
El BASIC del Dragon 32 está programado desde un nivel elemental hasta uno extenso para sonido y gráficos. Se completa con una serie de apéndices que incluyen prácticas plantillas. Éstas se usan para crear diseños gráficos antes de programar.
Existe una buena selección de libros referentes al Dragon 32, tanto sobre cursos de programación como de listado de programas

Consultando al chip

Los "sistemas especializados" liberarán de la rutina a los expertos; se trata de ordenadores programados para analizar información compleja y responder preguntas relativas a la misma

La inteligencia artificial, es decir, la creación de ordenadores que piensen y tomen decisiones del modo en que lo hacen sus creadores humanos, es aún ciencia-ficción. La comprensión total del cerebro humano y de su funcionamiento es una tarea colosal, y a pesar de los avances que se están produciendo, las posibilidades de llegar a crear un ordenador inteligente, como el que aparece en la película *2001: una odisea del espacio*, seguirán siendo remotas durante muchos años.



Cortesía de Stanfords Ltd.

Panorama desde el espacio
En julio de 1982 se lanzó el Landsat 4, que se desplaza en una órbita que cubre toda la superficie de la Tierra pasando por el mismo sitio una vez cada veinte días. Toda la información del sensor se envía en forma digital y, utilizando las técnicas informáticas, pueden resolverse objetos de sólo 40 metros de ancho y se pueden interpretar las características geográficas. En esta ilustración se ha procesado fotográficamente la información digital para mostrar aspectos de Londres y del sudeste de Inglaterra. Las aguas claras se ven en azul oscuro, y las aguas poco profundas, con sedimentos, en azul claro; las ciudades y los campos labrados se observan en azul grisáceo, los terrenos baldíos, en rojo amarronado; la cosecha de cereales tiene color verde y otras clases de vegetación se ven en rojo vivo

Pero si esta tarea pudiera limitarse, si el ordenador sólo hubiera de parecer "inteligente" en un campo muy restringido de la actividad humana, entonces sería mucho más sencillo reproducir, al menos, la apariencia de inteligencia.

Ésta es la teoría que sustenta a los sistemas especializados. El ideal perseguido concibe que un especialista en un campo determinado, por ejemplo, un geólogo o un experto cirujano, pudiera alimentar a un sistema de ordenador con sus conocimientos y con las reglas para aplicarlos. De esta manera, el programa de ordenador para manipular el conocimiento y las reglas quedaría a disposición de las personas no especializadas, quienes podrían digitar preguntas relativas a este campo del saber y recibir respuestas significativas.

Los sistemas especializados podrían ser de gran utilidad en diversos sentidos. Ya se ha desarrollado un programa para diagnosticar las causas del dolor de estómago interrogando a los pacientes acerca de sus síntomas. Otro programa se basa en el conocimiento humano de la geología para detectar los sitios donde existan mayores probabilidades de hallar molibdeno y otros minerales. Y un tercer programa deduce la probable estructura de moléculas orgánicas a partir de masas de información experimental no estructuradas. Normalmente todas estas tareas las asumirían científicos profesionales de altísima categoría cuya capacitación resulta sumamente onerosa. Gracias a los ordenadores, estas valiosas personas pueden ahora emprender trabajos más originales.

Pero los sistemas especializados tienen algo más que ofrecer aparte de la mera sustitución de los especialistas humanos. Una vez que el programa utiliza conocimiento especializado, el ordenador suele descubrir algunos hechos inesperados. Algunas veces la máquina detecta relaciones existentes entre un dato y otro que habían pasado inadvertidas para el hombre, y sugiere nuevas vías de investigación.

Existe, pues, la creencia generalizada de que los sistemas especializados suponen o, al menos, supondrán un importante avance en cuanto a las aplicaciones de los ordenadores. Si el programa a cargo del sistema funciona correctamente, el ordenador puede convertirse en un testigo especializado. Y muchos ordenadores pueden usar el mismo programa, traspasando el conocimiento especializado de una única persona a una gran cantidad de ordenadores igualmente especializados. El problema obvio con el que se enfrentan los investigadores es el de escribir un programa que *funcione* correctamente: un programa que sea tan "inteligente" como un especialista humano.

Creando el programa

El primer paso consiste en pensar de qué modo los expertos toman decisiones sobre la evidencia y las preguntas relativas a su especialidad. El pensamiento humano no es particularmente lógico, en especial si se lo compara con la forma en que trabajan los ordenadores, y depende en gran parte de la experiencia. Si a un especialista humano se le plantea una pregunta o un problema nuevos, los compara mentalmente con la gran cantidad de diversas situaciones con las que se ha enfrentado antes. Luego de comparar la nueva situación con aquellas que ya posee en su memoria, puede llegar a algunas conclusiones provisionales y emprender la acción adecuada.

Pero representar el conocimiento profundamente detallado que posee, por ejemplo, un médico acerca de estas estructuras, implica almacenar y relacionar de formas muy complejas una enorme cantidad de reglas. Y son necesarias aún más modificaciones para que el ordenador imite el comportamiento humano. Los médicos nunca están absolutamente seguros de casi nada y al emitir una opinión sólo pueden decir que están "casi seguros" o "confían mucho" en ella. Basándose en sólo un par de síntomas, un médico sólo podría estar seguro de su diagnóstico en un 30 %.

De modo que en nuestro modelo de ordenador las conclusiones deben contar con unos valores de proba-

bilidad que se sitúan entre el 100 %, cuando sólo hay una conclusión posible, y el 1 %, cuando existen otras cien conclusiones igualmente posibles.

La mayoría de los sistemas especializados trabajan estableciendo un diálogo entre el interrogador y el ordenador. El interrogador ha de dar entrada a los detalles del problema que debe resolver el ordenador especializado; y en este sentido, el modo más sencillo de evitar problemas consiste en hacer que el ordenador interrogue al programador mediante una serie de preguntas objetivas (con varias respuestas posibles). Con esto se evita el riesgo de que el usuario dé entrada a palabras u oraciones que el ordenador no pueda comprender. Luego el ordenador compara la información de entrada con las reglas del conocimiento que tiene almacenado. La ruta que siga el ordenador en esta red de reglas depende de las respuestas que el usuario dé para sus preguntas, y a medida que el programa avanza a través de esta red, cada paso determina la siguiente pregunta del ordenador.

El sistema especializado más conocido en Gran Bretaña, el programa Mickie para diagnóstico médico, se ejecuta en un microordenador. Y están en proceso de fabricación otros productos comerciales que convertirán al ordenador personal en un especialista en diversos temas. Cierto es que aún no podemos hablar con nuestros ordenadores, pero sí formularles preguntas y recibir respuestas fiables.

Una red de animales

En este diagrama vemos cómo se organiza el "conocimiento" en un sistema especializado, de acuerdo a una serie de reglas sencillas. En este caso las reglas se numeran de R1 a R15 y tienen el fin de identificar a un animal a partir de la información que se tiene sobre él. Por ejemplo, supongamos que se da entrada a la información de que el animal posee dientes puntiagudos. El ordenador comienza por el recuadro superior izquierdo del diagrama y mira qué camino de la red puede tomar para llegar hasta uno de los animales incluidos en la parte inferior. La palabra AND (y) en el interior del círculo significa que las tres condiciones de los recuadros que se dirigen a ella han de ser verdaderas antes de que se pueda aplicar R6. El ordenador no puede seguir adelante hasta que no se le proporcione más información y requiere que se le respondan dos preguntas: "¿tiene garras ese animal?" y "¿tiene los ojos en la parte anterior de la cabeza (ojos delanteros)?" Si la respuesta es positiva en ambos casos, R6 permite que el ordenador continúe hacia abajo. Pero R6 conduce a una OR (o), lo que significa que sólo podrá seguir adelante si R6 o R5 son verdaderas. En este caso R6 es verdadera, de modo que el ordenador continúa. Aquí hay dos posibilidades AND, cada una de las cuales lleva a conclusiones diferentes. Para aplicar ya sea R9 o R10, y dar la respuesta LEOPARDO o TIGRE, el ordenador ha de formular más preguntas. En este caso ha de preguntar si el animal posee franjas negras o manchas oscuras. Con esta información ya podrá dar una respuesta. Éste es el diagrama clásico de la forma en que funciona el juego por ordenador "Animales"

Julie-Ann Chambers/Mark Watkinson

Creadas para imprimir

La impresora margarita, la impresora de chorro de tinta y la impresora matricial son el más reciente avance en la tecnología de impresión, y se van introduciendo en la oficina y en el hogar

Guía del papel
La impresora se provee de papel mediante un rodillo rotatorio con púas que engancha las perforaciones situadas en el margen del papel

La rueda margarita
La rueda margarita lleva los caracteres acoplados en el extremo de sus "pétalos". Al pulsar una tecla la rueda se mueve hasta el pétalo adecuado para que se pueda imprimir el carácter

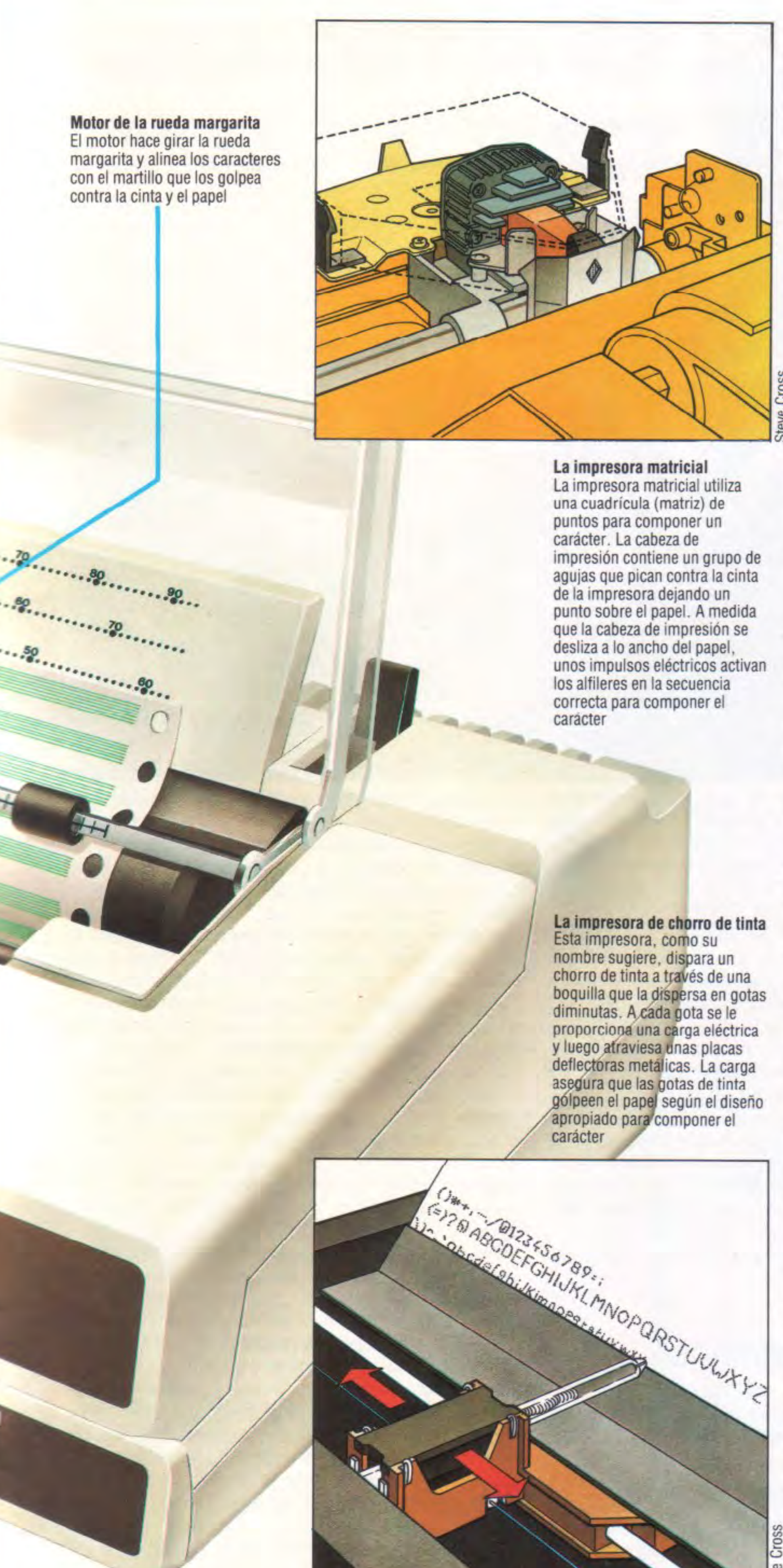
David Weeks

Cartucho de cinta de impresión
La mayoría de las impresoras margarita poseen un cartucho de cinta de impresión que puede cambiarse en cuestión de segundos. La nueva cinta viene en una caja plástica que se coloca en la impresora

El martillo
Un pequeño "martillo" de metal golpea el carácter en los extremos de la rueda margarita y lo impulsa sobre la cinta, estampando la impresión en el papel

Tensor de cinta
Haciendo girar el tensor la cinta se afloja o se ajusta. Se suele utilizar al cambiar la cinta, ya que a menudo la nueva se ha aflojado en el interior del cartucho antes de ser usada

Panel de control
Este panel posee dos mandos principales, que colocan a la impresora "en línea", es decir, la preparan para recibir información. También, si la impresora se detiene en mitad de una página, desplazan el papel hasta el inicio de la siguiente. Otro mando es el "avance de línea", que desplaza el papel hacia arriba una línea cada vez



Motor de la rueda margarita
El motor hace girar la rueda margarita y alinea los caracteres con el martillo que los golpea contra la cinta y el papel

La impresora matricial
La impresora matricial utiliza una cuadrícula (matriz) de puntos para componer un carácter. La cabeza de impresión contiene un grupo de agujas que pican contra la cinta de la impresora dejando un punto sobre el papel. A medida que la cabeza de impresión se desliza a lo ancho del papel, unos impulsos eléctricos activan los alfileres en la secuencia correcta para componer el carácter

La impresora de chorro de tinta
Esta impresora, como su nombre sugiere, dispara un chorro de tinta a través de una boquilla que la dispersa en gotas diminutas. A cada gota se le proporciona una carga eléctrica y luego atraviesa unas placas deflectoras metálicas. La carga asegura que las gotas de tinta golpeen el papel según el diseño apropiado para componer el carácter

Quizá hasta ahora no haya pensado seriamente en la posibilidad de usar una impresora. Después de todo, si está satisfecho utilizando su ordenador personal para jugar o para calcular los gastos de la familia, en realidad no necesita el texto impreso de lo que se visualiza en su monitor o en la pantalla de su televisor.

Pero cuando adquiera más experiencia en cuanto a la utilización de su ordenador personal, comprobará que trabajar seriamente careciendo de una impresora tiene unas limitaciones evidentes. Si le interesa escribir sus propios programas, querrá guardar copia de los listados de los mismos. Si utiliza su ordenador para llevar sus cuentas, necesitará un registro impreso de los cálculos.

La elección de la impresora adecuada a sus necesidades debe ser cuidadosa. Lo que haya de pagar por ella dependerá de la velocidad de la impresora en producir palabras y de la calidad de los resultados.

Elección de la impresora

Existen tres clases principales de impresoras para ordenadores personales: la matricial, la margarita y la térmica.

El método de impresión más difundido es el matricial. Éste consiste en una cabeza de impresión que contiene un grupo de agujas. Los caracteres se imprimen mediante las combinaciones de estas agujas al golpear contra la cinta. El método matricial tiene la ventaja de que trabaja muy rápido y de que las impresoras son relativamente económicas. Sin embargo, como las letras y los números se componen de una serie de puntos, la calidad de impresión suele ser muy pobre. Por otra parte, las impresoras de este tipo son bastante ruidosas.

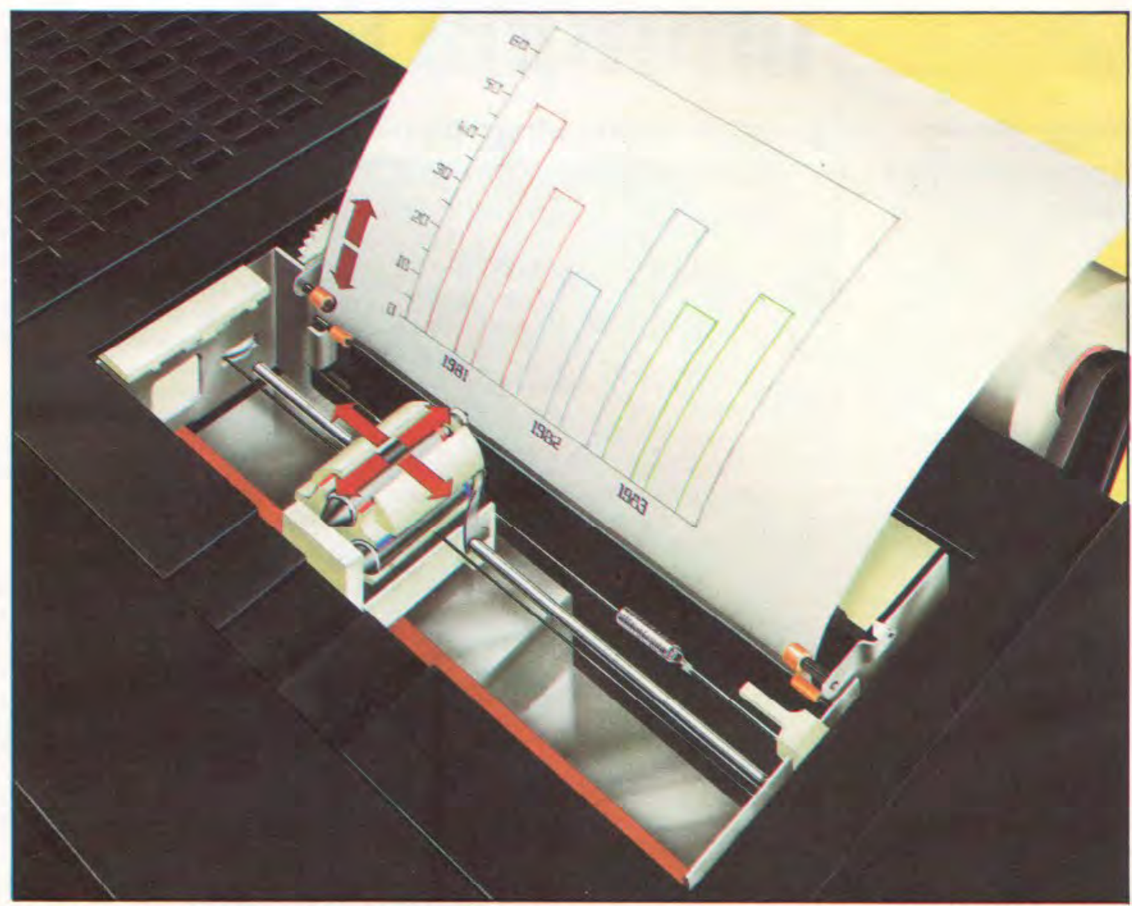
Algunas impresoras matriciales han superado el problema de la baja calidad de impresión sobreimprimiendo los puntos dos o tres veces. En este caso, la cabeza de impresión se desplaza ligeramente para que los nuevos puntos encajen entre los puntos impresos previamente.

Las impresoras matriciales son aceptables si usted desea guardar borradores de lo que ha producido su ordenador. También pueden realizar gráficos y diagramas, ya que a la máquina le es posible imprimir patrones además de caracteres. Pero necesitará una impresora diferente si lo que desea es una impresión de gran calidad para, por ejemplo, enviar una carta al gerente de un banco.

Para obtener una calidad similar a la de las máquinas de escribir, ha de optar por una impresora margarita, así llamada porque utiliza una rueda con largos "pétalos" de aspecto parecido a los de dicha flor. En el extremo de cada pétalo hay una letra, un símbolo o un número. Para imprimirlos, la rueda gira para alinear cada pétalo con un pequeño "martillo" de metal que impulsa sobre la cinta y el rodillo el carácter situado en el extremo del pétalo. La rueda margarita puede ser plástica o metálica. También puede sustituirse por otra provista de un tipo de impresión diferente para obtener distintas clases de impresión, al igual que si se tratara de una máquina de escribir cuyos tipos están dispuestos en una bola.

El principal inconveniente de las impresoras margarita es que su funcionamiento es mucho más lento que el de las impresoras matriciales; además, suelen ser más caras. Tampoco sirven para diagramas y gráficos, ya que para producir las diversas formas de un gráfico se necesitarían varias ruedas.

La impresora "esferográfica"
Representa un reciente avance en cuanto a técnicas de impresión se refiere. La cabeza de impresión sostiene cuatro bolígrafos (plumas esferográficas) diseñados especialmente. Cuando se da la orden PRINT, el papel se mueve hacia arriba y hacia abajo para producir los trazos verticales del carácter, mientras los bolígrafos se mueven hacia los lados para crear los trazos horizontales. La ventaja de este sistema reside en que puede utilizarse para imprimir diagramas y gráficos en colores. Por otra parte, ofrece una mejor calidad de impresión que el procedimiento matricial, ya que sus caracteres se componen sencillamente a partir de los trazos del bolígrafo. En contrapartida, este sistema es comparativamente lento y los bolígrafos han de reemplazarse con regularidad si se imprimen de forma continuada textos muy prolongados



Steve Cross

Las impresoras de chorro de tinta resultan aún más caras. Éstas disparan gotas de tinta para modelar el carácter que se ha de imprimir. La tinta es conducida a través de una boquilla que la dispersa en gotas diminutas. Estas gotas atraviesan luego un electrodo y reciben unas cargas eléctricas. Un par de placas metálicas desvían luego las gotas en distintas direcciones, para modelar la forma del carácter. ¡Las impresoras de chorro de tinta son tan rápidas que pueden imprimir alrededor de 20 metros de caracteres por segundo!

Un método alternativo, la impresora térmica, utiliza papel sensible al calor. La cabeza de impresión traspasa su calor al papel de modo tal que éste se vuelve negro en el área tocada, formando el carácter deseado. Las impresoras térmicas son muy silenciosas y bastante rápidas. Uno de los modelos más populares es el Apple Silent Type. El precio de las impresoras térmicas es bastante razonable; no obstante, con ellas necesitará utilizar un papel especial sensible al calor, más caro que el corriente, y, por otra parte, la calidad de impresión no es tan buena como la de las impresoras margarita.

La interface apropiada

Cuando decida comprarse una impresora, además de determinar exactamente para qué desea emplearla, en términos de calidad y velocidad de impresión, también habrá de asegurarse de que pueda utilizarla con su propio ordenador. La clavija conectora de la impresora debe ser compatible con el ordenador. El conector para enchufar la impresora está situado generalmente en la parte posterior del ordenador y se denomina "interface".

Los tres tipos de interface más corrientes son el Centronics, el IEEE488 y el RS232. El Centronics también se conoce como interface "paralela". Su ordenador ha de tener una abertura para al menos una de estas tres interfaces.

Sin embargo, la industria de la informática es notoriamente conocida por su incompatibilidad, y puede encontrarse con el hecho de que, aun utilizando la misma conexión, un ordenador y una impresora sean incompatibles. Esto se debe a que la interface debe trabajar a la misma velocidad tanto en el micro como en la impresora. La unidad para medir esta velocidad se denomina *baudio* y corresponde a la velocidad a la cual se transfieren a la impresora los bits de la memoria del ordenador. Los bits se pueden enviar a la impresora de dos maneras: transfiriéndolos uno detrás de otro a través de un único cable, como en la interface RS232 (procedimiento "en serie"), o bien transmitiéndolos simultáneamente a través de varios cables, como en las interfaces Centronics e IEEE488 (procedimiento "en paralelo").

Existen dos formas principales de proveer de papel a las impresoras: hoja por hoja, como en una máquina de escribir, o por "arrastre", mediante dos ruedas dentadas que sujetan el papel por unos agujeros perforados situados a ambos márgenes de la hoja; se trata más o menos del procedimiento que utilizan las cámaras fotográficas para hacer correr la película de una exposición a otra. El papel de arrastre o para ruedas dentadas ofrece la ventaja de que se puede dejar que la impresora se alimente a sí misma. Sin embargo, este método no admite la clase de papel con membrete que ha de alimentarse a la impresora de una hoja cada vez.

Al comprar una impresora, debe decidir para qué desea utilizarla y luego adquirir la más adecuada para su ordenador.

Cuestiones de rutina

Programas dentro de otros: presentamos un nuevo aspecto del Basic, que hará que los programas sean claros y manejables

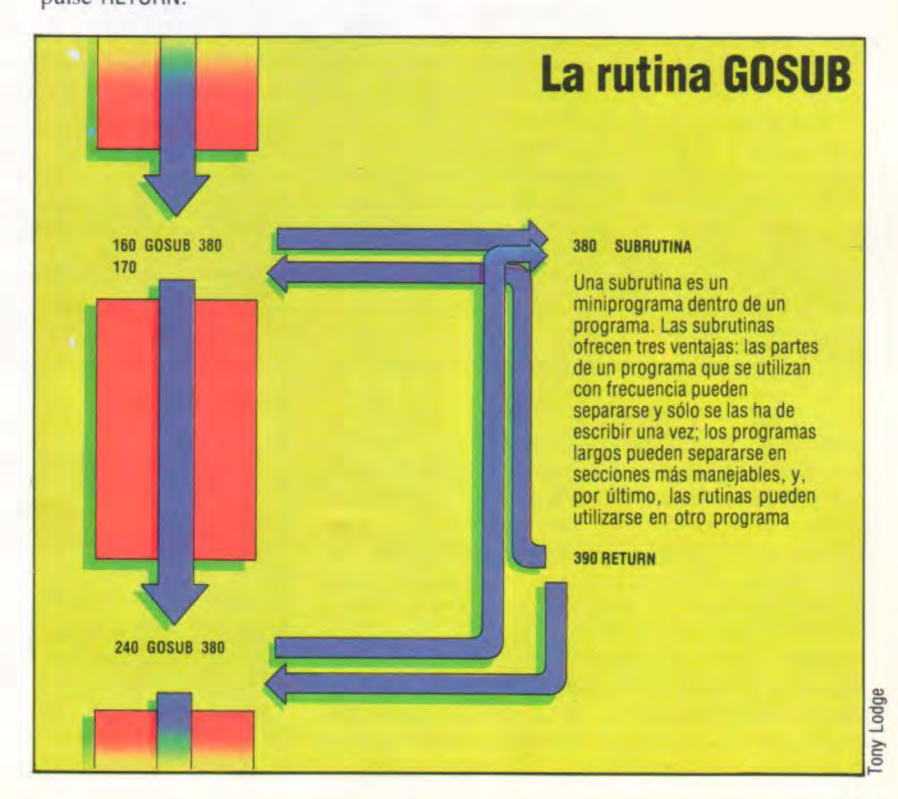
En anteriores entregas de nuestro curso de programación BASIC hemos digitado programas, los hemos ejecutado, hemos introducido modificaciones y luego borrado la memoria (mediante la orden NEW) cuando deseábamos dar entrada a nuevos programas. Cuando hemos necesitado volver a ejecutar otra vez el antiguo programa, ha sido necesario volver a digitarlo.

Para ahorrarnos este trabajo repetitivo, todas las versiones de BASIC se suministran con una orden que permite almacenar cualquier programa en cinta de cassette. El siguiente programa puede guardarse en cinta utilizando la orden SAVE seguida del nombre de archivo. El programa servirá para calcular el número de azulejos que se necesitan para revestir las paredes de una habitación.

```
10 REM ESTE PROGRAMA CALCULA EL NUMERO DE
   AZULEJOS
20 REM QUE SE NECESITAN PARA AZULEJAR UNA
   HABITACION
30 PRINT "DE ENTRADA A LA MEDIDA DE LOS LADOS
   DEL AZULEJO EN MM"
40 INPUT A1
50 REM LA LINEA 60 HALLA LA SUPERFICIE DEL
   AZULEJO
60 LET A2 = A1 * A1
70 PRINT "DE ENTRADA AL NUMERO DE PAREDES"
80 REM W ESTABLECE EL LIMITE DEL BUCLE
90 INPUT W
100 FOR X = 1 TO W
110 PRINT "LONGITUD DE LA PARED N.º"; X;
    "EN METROS"
120 REM D ES LA DIMENSION DE LA PARED
130 INPUT D
140 REM SE CONVIERTE EN MM
150 REM EN LA SUBRUTINA
160 GOSUB 380
170 REM LINEA 190 ESTABLECE L EN
180 REM LONGITUD DE LA PARED EN MM
190 LET L = D2
200 PRINT "ALTURA DE LA PARED N.º"; X;
    "EN METROS"
210 REM LINEAS 230 HASTA 250 ESTABLECEN H
220 REM EN LA ALTURA DE LA PARED EN MM
230 INPUT D
240 GOSUB 380
250 LET H = D2
260 REM LINEA 270 ESTABLECE QUE A3 ES LA
    SUPERFICIE DE LA PARED
270 LET A3 = L * H
280 REM S (SUBTOTAL) ES LA SUPERFICIE
    DE LA PARED DIVIDIDA
290 REM POR LA SUPERFICIE DEL AZULEJO
300 LET S = A3/A2
310 REM T (TOTAL) TIENE EL NUEVO SUBTOTAL
320 REM AGREGADO CADA VEZ QUE REALIZA
    EL BUCLE
330 LET T = T + S
340 NEXT X
350 REM PRINT EL TOTAL
360 PRINT T
370 END
380 LET D2 = D * 1000
390 RETURN
```

Habiendo digitado el programa, todo lo que ha de hacer para guardarlo en la cinta de cassette es utilizar la orden SAVE. Antes, por supuesto, debe preparar el ordenador de acuerdo a las instrucciones del manual. La orden SAVE es muy sencilla de utilizar. Basta con que digite SAVE seguido de un nombre de archivo entre comillas dobles. Un nombre de archivo es la denominación que se le da a un archivo y, en la terminología de informática, equivale a un programa o una serie de datos que pueden almacenarse o recuperarse cuando se desee. Lo mejor es utilizar un nombre de archivo que le recuerde la función del programa. Dado que nuestro programa calcula el número de azulejos que se necesitan para azulejar una habitación, podríamos denominarlo "AZULEJOS". Una vez preparada la grabadora de cassette, introduzca en ella una cinta virgen para retener el programa.

Las grabadoras de cassette con conector para control remoto pueden controlarse, por lo general, directamente mediante el ordenador. De lo contrario, coloque la grabadora en la modalidad *record* y luego dispóngala en la modalidad *pause*. Digite la orden SAVE, incluyendo el nombre de archivo. Ponga en funcionamiento la grabadora soltando el mando *pause* y luego pulse RETURN.



Una subrutina es un miniprograma dentro de un programa. Las subrutinas ofrecen tres ventajas: las partes de un programa que se utilizan con frecuencia pueden separarse y sólo se las ha de escribir una vez; los programas largos pueden separarse en secciones más manejables, y, por último, las rutinas pueden utilizarse en otro programa

Tony Lodge

Para comprobar si el programa se ha grabado correctamente, borre la memoria del ordenador digitando NEW <CR>. Rebobine la cassette, coloque la grabadora en la modalidad *play* y luego vuelva a cargar el programa en el ordenador utilizando la orden LOAD. LOAD debe ir seguida del nombre de archivo del archivo deseado. Digite LOAD "AZULEJOS" y enseguida pulse RETURN.

Luego de que el programa haya sido cargado en el ordenador, un mensaje en la pantalla tal como READY u OK indicará que se ha completado la carga. Liste el programa (LIST) y verifique que se trata del mismo que ha digitado.

GOSUB

GOSUB es una sentencia que desvía el flujo de un programa a una subrutina. Una subrutina es como un miniprograma separado o un programa dentro de otro programa. En el programa que hemos empleado para ilustrar el tema, la subrutina es muy sencilla. Se incluye para mostrar el principio, puesto que podrían haberse elaborado fácilmente otras maneras de producir los mismos resultados sin recurrir a la utilización de una subrutina.

Nuestro programa calcula el número de piezas que se necesitan para azulejar un cuarto, averiguando la superficie de los azulejos empleados. Luego solicita que se dé entrada a la longitud y la altura de cada una de las paredes. Calcula la superficie de la pared después de haber convertido la longitud y la altura de metros a milímetros. El número de azulejos necesarios se calcula dividiendo la superficie de cada una de las paredes por la superficie de un azulejo y luego sumando los resultados. La conversión de la longitud y la altura de las paredes a milímetros se realiza en la subrutina, que simplemente multiplica la longitud o la altura (en metros) por 1 000 para hallar el equivalente en milímetros.

Las subrutinas ofrecen tres ventajas: las partes de los programas utilizadas con frecuencia se pueden separar y sólo necesitan escribirse una vez, independientemente de cuántas veces se requiera la operación; permiten dividir programas largos y complicados en unidades o secciones más manejables y más fáciles de comprender; por último, las subrutinas pueden volver a usarse en cualquier programa que admita su función.

En nuestro programa, la subrutina empieza en la línea 380 y consiste en una sola sentencia: LET D2 = D * 1000. Toma a D, la dimensión de la pared (longitud o altura), y la multiplica por 1000 para convertirla de metros a milímetros. Al resultado se le asigna la variable D2.

La instrucción que obliga al programa a dirigirse a la subrutina es GOSUB. Se produce por primera vez en la línea 160. A la variable D se le había asignado el valor de la longitud de la pared en la línea 130. La línea 160 obliga al programa a dirigirse hasta la subrutina, donde a la variable D2 se le da el valor de D multiplicado por 1000. La instrucción RETURN en la línea 390 es necesaria para que el programa retorne desde la subrutina hasta el programa principal. Las subrutinas siempre vuelven a la línea que sigue a la sentencia GOSUB; en este caso, a la línea 170.

GOSUB vuelve a producirse en la línea 240, que "llama" a la misma subrutina. Esta vez, la subrutina regresa (RETURN) a la línea 250. Si bien este programa emplea una sola subrutina, se pueden utilizar cuantas se necesiten. En todos los casos la sentencia GOSUB ha de incluir el número de línea de la subrutina apropiada. Observe que la sentencia END se produce en la línea 370, antes que la subrutina. END indica el final del programa principal y también sirve para detener el programa, evitando que siga ejecutando las subrutinas después de haber finalizado.

A pesar de que este programa es algo más largo que los anteriores, no por ello es más complicado. Pruebe con este programa, sígalo entero, línea por línea, y vea lo que sucede en cada etapa. Además de la orden GOSUB y de las subrutinas, este programa sólo introduce otro concepto: nombres de variables más largos.

Le sería de gran ayuda dibujar recuadros con los nombres de las variables en su interior y escribir, además, los valores para cada etapa.

Línea 300: LET S = A3/A2 algunas veces dará un número con fracción decimal. Intente ejecutar el programa dando entrada a 110 mm como medida de los azulejos, y utilizando una sola pared de 2,3 y 1,8 metros de longitud y altura respectivamente. Debe obtener como respuesta 342,149 azulejos. Como los azulejos únicamente se venden por unidades enteras, esta respuesta no es del todo apropiada. En una próxima ocasión estudiaremos una de las maneras de conseguir una respuesta adecuada en números enteros.

Ejercicios

- Vea qué sucede si da entrada a 0 mm como medida del azulejo. Al final de la ejecución debería obtener un mensaje de error. ¿Y por qué? ¿Por qué no obtiene un mensaje de error similar cuando da una entrada de 0 metros para la longitud de una de las paredes? Una pista: no es igual multiplicar por cero que dividir por cero; ¡pruébelo en su calculadora!
- El programa sólo funciona cuando se trata de azulejos cuadrados. Vea si puede modificar desde la línea 30 hasta la línea 60 para hallar la superficie de azulejos rectangulares (del mismo modo que, más adelante en el programa, hemos hallado la superficie de paredes rectangulares).
- Agregue una sentencia en la línea 355 para aumentar el número total de azulejos en un 5 %, con el fin de compensar los azulejos que se estropeen. El aumento del 5 % se conseguirá multiplicando un número por 105/100.

Complementos al BASIC



El Spectrum no dispone de esta orden, lo que significa que se necesita otro procedimiento para pasar por alto las líneas 380 y 390. Éste consiste en modificar la línea 370 para que quede: 370 GOTO 400, y agregando 400 PRINT "END"



El Spectrum requiere que se definan todas las variables antes de que pueda realizar una operación aritmética. De modo que para que la línea 330 funcione correctamente, debe agregarse una nueva línea:
5 LET T = 0



En el Spectrum aparece como dos palabras separadas, si bien sólo se necesita una tecla

Cuando menos = más

Los ordenadores prefieren la sencillez, y por ello emplean un inteligente truco para efectuar una resta mediante una suma



El Voyager 2
El espectacular Voyager se convirtió en el primer explorador espacial que fue más allá de nuestro sistema solar. Viajó a través del espacio, tomando fotografías y acumulando información, mientras el ordenador de a bordo convertía los datos en dígitos binarios. La información se enviaba a la Tierra a la asombrosa velocidad de 116 000 bits por segundo. En la Tierra era procesada por los ordenadores de la NASA, en Houston (Texas)

Cortesía de the Science Photo Library

En la primera parte de esta serie descubrimos que se pueden emplear dígitos binarios para representar cualquier número decimal. Los números binarios tienen la desventaja de que son más largos que sus equivalentes decimales, pero son convenientes para el ordenador porque los ceros y los unos pueden representarse mediante voltajes negativos y voltajes positivos. También hemos visto que es muy sencillo sumar números binarios entre sí.

En el papel, los números binarios pueden restarse con igual sencillez que los números decimales, siguiendo las mismas reglas que se aplican a la resta en el sistema decimal. Sin embargo, hace tiempo que los diseñadores de ordenadores comprendieron que agregando circuitos (circuitos electrónicos para sumar) podían tanto sumar como restar sin necesidad de circuitos especiales de resta. Veamos cómo se realiza.

El complemento a dos

Un procedimiento para representar los números negativos en los ordenadores se denomina "complemento a dos". Con él, el proceso de la resta aparece como un aspecto más de la suma. Consideremos el siguiente problema aritmético:

$$16 - 12 = 4$$
$$\text{o } 16 + (-12) = 4$$

Aquí 12 se resta a 16, pero el proceso de sustracción puede considerarse, igualmente, como una adición: la suma de 16 y 12 negativo. En ambos casos la respuesta es la misma y la única diferencia es la utilización de signos aritméticos y de los paréntesis. Esta ligera modificación puede utilizarla el ordenador tanto para representar números negativos como para simplificar el problema de la resta.

Por razones de claridad, supongamos que la capacidad de nuestro ordenador sólo alcanza para manipular 5 dígitos. Por supuesto, los ordenadores reales pueden registrar números con miles de dígitos. Nuestro ordenador de 5 dígitos adopta un método de trabajo: el primer dígito por el lado izquierdo se considera separadamente de los otros 4. Si el dígito inicial es 1, representa 16 negativo; y si es 0, representa, por supuesto, un cero. Los cuatro dígitos restantes son positivos y responden a las convenciones binarias que hemos visto en el primer capítulo de este curso:

[ ]	[ ]	[ ]	[ ]	[ ]
−16 o 0	8	4	2	1 o 0

De manera que, por ejemplo, el número binario 01000 es el decimal 8 y 10000 es el decimal −16. Pero, ¿qué sucede con 10100? Éste incluye −16 y +4, dando −12.

¿Cuántos números pueden representarse con sólo 5 dígitos utilizando esta convención? El mayor número positivo es 01111 o decimal 15, y el mayor número negativo es 10000 o −16. Experimentando un poco verá que se pueden representar todos los números comprendidos entre −16 y +15.

Binario	**Decimal**
10000	−16
10001	−15
10010	−14
10011	−13
10100	−12
etc.	
11111	−1
00000	0
00001	1
00010	2
etc.	
01110	14
01111	15

Si aumentáramos el número de dígitos que pudiera manipular nuestro ordenador, ampliaríamos, por supuesto, la gama de números.

Ya en los primeros tiempos del desarrollo de la aritmética binaria para ordenadores, se descubrió un truco muy sencillo para hallar el complemento a dos, o forma negativa, de un número. Para llegar a este truco hay que seguir dos pasos.

Primero, invertir cada dígito. De modo que cada vez que encuentre un 1 ponga un 0, y cada vez que haya un 0 lo cambie por un 1. En segundo lugar, sumarle 1 al número invertido.

Siga el método tal como se expone en el ejemplo siguiente. Estamos utilizando +12, siendo 01100 su equivalente binario. (El 0 inicial sobre el lado izquierdo no es estrictamente necesario, puesto que 01100 es lo mismo que 1100. Pero, dado que nuestro ordenador posee 5 dígitos, debemos utilizarlos todos.)

	01100 (= +12)
Primer paso:	10011
Segundo paso:	00001 (+1)
	10100 (= −12)

Ahora veamos cómo aborda nuestro ordenador el problema de la resta, por ejemplo, 12 menos 4, utilizando el complemento a dos.

+12 es	01100
−4 es	11100 (usando el complemento a dos)
12 + (−4)	101000

Observe que ahora tenemos 6 dígitos. Puesto que la capacidad de nuestro ordenador sólo alcanza para registrar 5 dígitos, el primer dígito a la izquierda se denomina dígito de capacidad excedida y se ignora, ¡dejando 01000 u 8 decimal, que es la respuesta correcta! Veamos ahora un ejemplo algo más complicado: 4 menos 12.

+4 es	00100
−12 es	10100
4+(−12)	11000

Como ejemplo final, intentemos trabajar con dos números negativos al mismo tiempo: −3 −4 = = −3 + (−4) = −7

3 es	00011
por tanto, −3 es	11101 (usando el complemento a dos)
y −4 es	11100
	111001

Como podrá observar, nuevamente obtenemos un número de 6 dígitos. Una vez descartada la capacidad excedida, tenemos el número binario 11001 o −7 en decimal.

Estas restas utilizan sólo la suma y el truco del complemento a dos (que en sí mismo consiste sólo en la inversión de los dígitos y la suma). La ventaja para el ordenador es que los dígitos binarios pueden invertirse fácilmente empleando una puerta NOT (véase página 68).

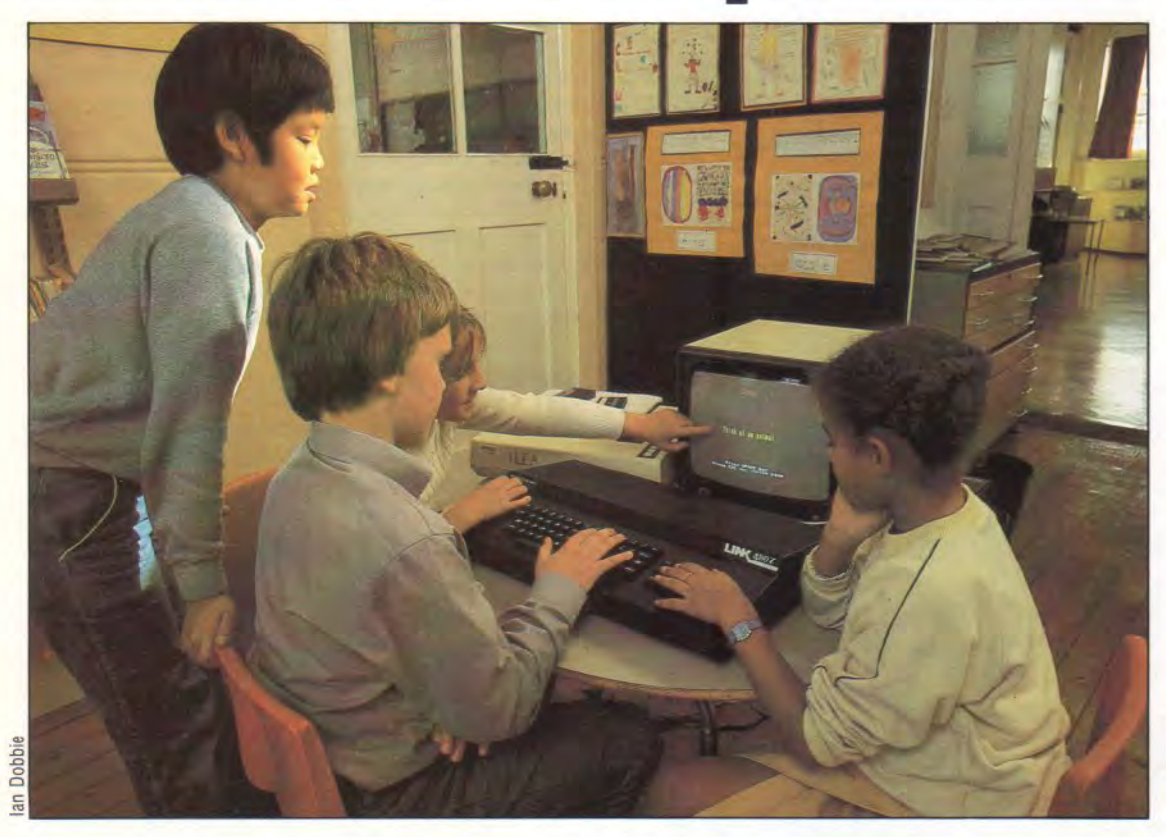
Una puerta NOT tiene una entrada y una salida. Es una puerta muy "díscola" porque cualquiera sea el valor que usted alimente, la salida será lo contrario. De manera que si la entrada es 0 la salida es 1, y si la entrada es 1 la salida es 0. Esta característica de "inversión" es exactamente lo que se necesita para el primer paso (la inversión) del truco del complemento a dos.

En el próximo capítulo de nuestra obra podremos ver cómo un ordenador tiene capacidad para sumar fácilmente utilizando una combinación de puertas lógicas.



Puntos y rayas
El código Morse es uno de los primeros ejemplos de la codificación binaria en la electrónica. En 1837 se instaló en Londres el primer telégrafo, con dos millas de cable que unían las estaciones de ferrocarril de Euston y Camden Town. Posteriormente, en ese mismo año, Samuel Morse mostró en los Estados Unidos su celebrado código para transmitir mensajes. Cada letra era una combinación de dos señales: puntos y rayas

Escuela en la pantalla



Ian Dobbie

El ordenador en la enseñanza básica
Diversos países se están preparando para el futuro mediante la aplicación de una política educativa que tiende a familiarizar a los niños con los ordenadores ya durante sus primeros años de escolarización. No se trata tan sólo de familiarizar al niño con la nueva tecnología, sino también de utilizar el ordenador para enseñar una gran variedad de temas, desde biología hasta lenguas extranjeras. Un ordenador es un maestro paciente e ideal, ya que no introduce un nuevo concepto hasta que el niño no domine por completo el problema anterior, permitiendo que tanto el alumno de comprensión lenta como el más despierto aprendan al ritmo de sus propias posibilidades. Como medio auxiliar de aprendizaje, el ordenador ofrece una gama de aplicaciones más amplias: por el solo hecho de utilizarlo, el niño descubre cómo se analiza y se resuelve un problema

La era de los ordenadores ya ha comenzado en las aulas, y existe en el mercado una fascinante y variada gama de programas educativos para niños de todas las edades

A finales de 1983 había en España más de 4 000 ordenadores personales dedicados a la enseñanza, y se estima que dentro de pocos años la mayoría de los centros de educación primaria y secundaria, así como también los de formación profesional, contarán al menos con un microordenador. En la actualidad los ordenadores se utilizan no sólo para impartir "nociones de informática", sino también, en el campo educativo, para enseñar temas relacionados con las matemáticas y la alfabetización, ayudar a los niños de comprensión lenta y estudiar lenguas extranjeras.

En el mercado existen muchos programas educativos para ordenadores personales, pero los maestros suelen quejarse de su poca calidad. Ello se debe a que son muy pocos los programas que se han escrito respetando por igual tanto la disciplina educativa como la informática.

Es muy raro que un programador de ordenadores posea experiencia en el campo de la enseñanza; y los maestros, muchos de los cuales apenas iniciados en este campo, incurren a veces en los más garrafales errores de programación.

Aunque lo más probable es que el programa de un maestro sólo se aplique en su propia clase, los problemas surgen en cuanto dicho programa se envía a otra escuela. Por lo general, el programa en sí mismo, almacenado ya sea en cassette o en disco, no es suficiente, haciéndose imprescindible una buena documentación explicativa, pues sin ella es posible que los estudiantes sean incapaces de hacer funcionar el programa. Ante este problema, tal vez la respuesta del programador fuera: "¡Por supuesto, *se entiende* que se ha de digitar LOAD (cargar)!", pero a quien no sabe nada acerca de los ordenadores se le han de explicar uno por uno todos estos detalles.

Pasando a un plano más concreto, una buena programación requiere anticiparse a todos los errores que pueda cometer un principiante. Esto es muy importante para asegurar que el programa sea, efectivamente, un buen medio auxiliar de enseñanza. Una buena programación consiste en algo más que eliminar de un programa todos los posibles márgenes de error, hasta el punto de que sólo ejecute lo que debería hacer exclusivamente cuando se pulse la tecla adecuada. La buena programación también debe asegurar que el programa no hará nada que no deba realizar cuando se pulse una tecla equivocada. Ésta es la parte más complicada de la escritura de un programa. Éste ha de ser capaz de recuperarse de los más flagrantes errores en que pueda incurrir un niño y, al mismo tiempo, inculcarle la idea de que usar un ordenador es algo sencillo y divertido.

A pesar de estos problemas, existe una amplia gama de programas educativos tanto para la escuela como

para el hogar. Un ordenador es una maravillosa herramienta educativa y, para acertar en la elección del software más adecuado para sus hijos, le será de gran utilidad conocer las diversas maneras en que se puede emplear.

Un ordenador puede utilizarse para instruir al niño en prácticamente cualquier tema. Si el programa es bueno, lo más probable es que el niño se sienta fascinado y muy motivado para aprender.

La clase de programas educativos más generalizada se conoce como "de procedimiento correcto y ejercicios". En un programa de este tipo, al niño se le proporcionan ejemplos y luego se le invita a resolver problemas similares. Normalmente el programa lleva la cuenta de los fallos y los aciertos del niño. Incluso estimula al alumno cuando éste da una respuesta correcta y le sugiere amablemente "inténtalo otra vez" cuando la respuesta es equivocada.

Para decidir qué programas son los más adecuados para su hijo, deberá tener en consideración diversos factores: la edad del niño, el modelo de ordenador que usted haya adquirido y lo que su hijo esté estudiando en la escuela.

Si aún no ha comprado un ordenador personal, pero piensa que cuando lo haga una de las funciones a que lo destinará en su hogar será la educación, valdrá la pena averiguar qué tipo de ordenador usa su hijo en la escuela. Si usted está en condiciones de comprar un modelo similar, su hijo podrá realizar en casa los mismos programas educativos que utiliza en el colegio. A muchas escuelas les interesa proporcionar a los padres copias de los programas que se emplean en clase, y estos "deberes" pueden representar una considerable ayuda. Si usted ya ha adquirido un ordenador personal y éste no es compatible con los ordenadores de la escuela, no se preocupe; la experiencia que adquiera su hijo al digitar ordenadores diferentes también será muy valiosa.

Es natural que la mayoría de los programas educativos existentes hayan sido creados para las marcas de ordenadores más competitivas del mercado, pero muchos fabricantes están potenciando el papel docente del ordenador. Existe una gama particularmente amplia de programas educativos para el Apple, el Commodore PET, el Tandy, el BBC Micro, Sinclair y Texas Instruments; no obstante, algunos de los fabricantes de software incorporados recientemente a este mercado aún deben aportar una oferta de programas realmente más variada.

Los programas educativos para cualquier ordenador pueden conseguirse a través de la empresa fabricante o por medio de diversas firmas productoras de software independientes. Estas últimas escriben programas para ordenadores y anuncian sus productos en las revistas de informática y en los establecimientos que venden ordenadores personales.

La elección adecuada

Los programas destinados a niños menores de ocho años se centran en la enseñanza de las operaciones aritméticas básicas y en la alfabetización. Una de las gamas más interesantes de programas educativos para niños pequeños es la que produce Texas Instruments. El ordenador personal TI-99/4A, por ejemplo, fue muy bien acogido por los padres norteamericanos y británicos, pues contaba con una amplia gama de programas educativos de TI producida tanto por Texas Instruments como por Scott, Foresman & Co., de Estados Unidos. El 99/4A es un ordenador de 16 bits, lo que significa que probablemente los programas escritos en código de lenguaje máquina sean mucho mejores que los programas escritos para la mayoría de los ordenadores personales corrientes de 8 bits.

Esto lo demuestran algunos programas de TI tales como *Beginning grammar* (Iniciación a la gramática), *Addition and subtraction* (Suma y resta) y *Number magic* (La magia de los números). Estos programas van almacenados en un cartucho plástico que se introduce en el TI-99/4A, por lo cual a los niños les resultan muy manejables. No obstante, el hecho de que el software sea norteamericano representa un problema, y



El mejor compañero
El microordenador ayuda a resolver los "deberes", acompaña en los ratos de ocio y proporciona amigos. Pero para obtener un buen rendimiento de esta maravillosa máquina, es necesario acertar en la elección de los programas adecuados de acuerdo con la edad de los niños

Cortesía de Investrónica, S.A.

algunos de los programas suponen un gran inconveniente para los maestros de otros países. Sin embargo, TI posee un magnífico ejemplo de LOGO (véase p. 34), si bien realmente se encuadra en la segunda categoría de herramientas de descubrimiento.

Para niños de más edad se cuenta con un software muy rico y variado. Los programas para edades entre 8 y 11 años se caracterizan por su complejidad y calidad, y la mayoría de ellos tienen como objetivo reforzar los conocimientos básicos del alumno y poner a prueba sus aptitudes. A esta edad el niño comienza a interesarse por otros temas, como la música y las lenguas extranjeras, que pueden enseñarse, asimismo, por ordenador. Existen programas de esta naturaleza para la mayoría de los aparatos.

Para alumnos de enseñanza media existe una multitud de programas. Para hacer una selección de ellos y escoger sólo los mejores, lo más indicado es que hable con el profesor de su hijo.

Es muy importante que el trabajo escolar que realice el niño en su casa siga la misma línea que el que lleva a cabo en la escuela; la mayoría de los profesores se mostrarán bien dispuestos para orientar y ayudar a los padres en la elección del tipo de software educativo más apropiado para sus hijos.

Existe aún otra categoría de programas educativos para ordenador, destinada fundamentalmente a los niños menores de 13 años. A esta edad los niños aún están descubriendo la forma de aprender y para ello resultan muy valiosos los programas que los introducen en el uso del ordenador como medio de descubrir el mundo por sí mismos. El programa más conocido de esta categoría es el LOGO, lenguaje del cual existen versiones para ordenadores de las firmas Atari, Tandy, Apple, Texas Instruments, Research Machines, Commodore e IBM. Por su parte, Sinclair y BBC han mostrado interés, también, por realizar sus versiones, pero aún no se encuentran en el mercado. A través de este programa, el niño entre 6 y 12 años de edad es estimulado para explorar el potencial gráfico del ordenador (y, a su vez, la geometría), mediante una "tortuga". El niño descubre cómo enseñarle a la tortuga a recordar procedimientos (programas) y, en determinados casos, puede ir progresando hasta llegar a dibujar en la pantalla todo un mundo de fantasía. Al utilizar estos programas, lo que el niño hace realmente es enseñarse a sí mismo las leyes básicas de las matemáticas y, en este sentido, se ha reconocido claramente la efectividad de este programa para ayudar a comprender conceptos matemáticos y espaciales.

Escoger un buen programa educativo no es una labor fácil, debido a la gran cantidad de software disponible. Es muy interesante asistir a alguna de las ferias de informática que se celebran periódicamente en el país. Con toda probabilidad, allí se encontrará con los propios fabricantes y programadores exhibiendo sus productos y, de este modo, tendrá la ocasión de conocer, probar y comparar programas.

Existe la opinión generalizada de que la actual escasez de buenos programas, que satisfagan tanto las exigencias relativas al ordenador como a la educación en sí misma, no ha de durar mucho tiempo más. Todos los meses aparecen programas nuevos que, con toda seguridad, serán un valioso estímulo para el desarrollo intelectual de sus hijos. Conviene tener presente a este respecto que, en un próximo futuro, la informática será indispensable en todos los campos de la actividad humana.

El Agente Secreto

USTED ES UN AGENTE DE MI6

SU MISIÓN ES CAPTURAR A UN FAMOSO ESPÍA

OBJETIVO ESPÍA: ELIMINAR SUS AGENTES EN CADA CIUDAD

Actualmente existen muchos programas educativos en el mercado. El que muestran las fotografías se denomina *Agente Secreto* y ha sido realizado por Heinemann. Su misión consiste en capturar a un famoso espía enemigo que se halla en Europa, antes de que él elimine a todos sus agentes. Para poder atraparlo, se han de descifrar las claves acerca de su paradero. Comienza la búsqueda...



Al espía sólo se le puede capturar en una ciudad; pero él no permanece en el mismo sitio durante más de dos horas. Cuando cree saber dónde se halla éste, usted tiene la alternativa de dirigirse hasta allí en tren o en avión. Deberá decidir qué es más importante: llegar rápido o el precio a pagar



Cada vez que alguno de los agentes que trabajan para usted le envíe un mensaje, se encenderá una luz intermitente en el mapa que le indicará desde qué ciudad proviene. Si su agente fuera eliminado, el mensaje será interceptado antes de que usted lo reciba. En ese caso, tal vez le interese contratar los servicios de un nuevo agente; pero entonces habrá de pagar por él...



Los informadores estarán felices de venderle información, pero ésta se envía en clave, por lo cual quizá le interese entregársela a un especialista para que le ayude a descifrarla. Al final del juego habrá aprendido los nombres y la situación de las más importantes ciudades europeas, entenderá los anuncios de salidas y llegadas en estaciones y aeropuertos y será un experto en la confección de presupuestos

Ian McKinnell

Descifrando el código

Digite su lenguaje de ordenador en el teclado y un programa que funciona dentro del micro lo convertirá rápidamente en su propio código de lenguaje máquina

A pesar de que parece que todos los microordenadores desempeñan funciones similares, cada modelo es exclusivo. Algunos se venden con programas ya incorporados, mientras que otros requieren que estos programas se les "lean" desde un disco o una cinta de cassette exteriores.

Algunas máquinas poseen un único programa global que permite tanto la entrada de programas como la utilización de órdenes directas como SAVE o LOAD. Para realizar estas dos funciones, otros modelos necesitan programas separados.

No obstante, la mayoría de los microordenadores más conocidos funcionan de acuerdo a principios similares. El movimiento de la información hacia y desde dispositivos de almacenamiento externos (discos y cintas) a la pantalla se controla, en ambos casos, mediante el teclado. Asimismo, todas las máquinas pueden comunicarse con otros dispositivos externos, como impresoras, trazadores de gráficos e instrumentos científicos. Y la mayoría de los micros admiten que sus usuarios escriban programas en lenguajes similares al corriente, como el BASIC.

Cuando digita en el teclado de un ordenador un programa en BASIC, otro programa denominado *sistema operativo* pasa lo que ha digitado tanto a la pantalla como a un programa interpretador de BASIC. Esto significa que en el interior del ordenador se están llevando tres programas simultáneamente: el sistema operativo, el interpretador de BASIC y su propio programa.

Cuando usted ejecuta su programa, estos tres programas deben actuar al mismo tiempo. Cada instrucción en BASIC de su programa es traducida por el interpretador, y las instrucciones resultantes en código de lenguaje máquina pasan, una por una, al microprocesador para que las ejecute. Al mismo tiempo, el sistema operativo está explorando el teclado para detectar la entrada de datos y poder visualizarlos en pantalla.

Si una de las instrucciones de su programa solicita que algo se imprima o se grabe en disco, por ejemplo, el interpretador le requeriría al sistema operativo que realice esta tarea.

La ilusión de que en el mismo momento se están produciendo diversas acciones se crea por la increíble velocidad con que trabaja el microprocesador. Éste procesa las instrucciones del sistema operativo y del interpretador con tanta rapidez que ambas pueden ejecutarse simultáneamente.

Algunas máquinas trabajan a velocidad aún mayor, permitiendo la llegada de datos para "interrumpir" el procesamiento normal. De esta forma, no es necesario que el sistema operativo verifique las diversas fuentes de entradas de datos externos, como el teclado o las unidades de disco.

Se denomina *editor* a un tipo de procesador de textos menos sofisticado. La calidad de los editores suele variar considerablemente; no obstante, es probable que el editor incorporado del interpretador de BASIC sea de similar calidad.

Para ejecutar sus programas en BASIC, en lugar de emplear un interpretador puede utilizar un compilador. Mientras que el interpretador ha de traducir una instrucción cada vez que se le proporcione, el compilador traduce todo su programa, completo, al código de lenguaje máquina del ordenador una sola vez y para siempre. Los programas que han sido "compilados" se ejecutan mucho más rápidamente que el software que ha sido "interpretado".

El BASIC es ideal para escribir programas. Ofrece la ventaja de que es muy similar al lenguaje corriente y muy apropiado para ser utilizado por los principiantes. Pero el programador más experimentado podrá ejecutar sus programas con mayor rapidez utilizando un lenguaje ensamblador. Éste no se parece en absoluto al lenguaje común y, por lo tanto, el programador ha de poseer un conocimiento bastante detallado de cómo realiza sus funciones el microprocesador.

Todas las instrucciones que se le dan al ordenador poseen un equivalente directo en código de lenguaje máquina. Un lenguaje ensamblador está constituido por una serie de instrucciones abreviadas, como MVI *(Move Immediate)* o JZ *(Jump on Zero)*. Éstas se usan para ayudar al programador a recordar sus funciones.

Si el usuario domina un lenguaje ensamblador, lo único que le quedaría por aprender sería el código de lenguaje máquina, pero ello no tendría mucho sentido, a menos que verdaderamente necesitara ganar mínimas fracciones de segundo en el tiempo de ejecución de un programa.

Los códigos de lenguaje máquina de los microordenadores generalmente se escriben de una forma denominada *hexadecimal*. Se trata de una forma de numeración de base 16. Se cuenta de 0 a 9 normalmente, y



El sistema operativo de disco
Cuando los programas se almacenan en un disco flexible, la información se distribuye al azar alrededor de la superficie del disco. El sistema operativo de disco es un programa que automáticamente lleva el registro de la localización de cada byte de información. La ilustración muestra la información almacenada en una sección pequeña de un disco. En el bloque izquierdo, ésta se representa en forma hexadecimal y, en el bloque de la derecha, figura el carácter equivalente. Los códigos que no corresponden a caracteres que han de imprimirse aparecen en forma de puntos

luego se continúa utilizando las letras del alfabeto desde la A a la F para los números del 10 al 15. Cada dirección de memoria comprende ocho dígitos binarios (bits) y éstos se pueden representar mediante un par de dígitos hexadecimales.

Por ejemplo, el número binario 01011101 se ha de dividir primero en dos mitades: 0101 y 1101. Éstas se traducirían en los números decimales 5 y 13, que en el sistema hexadecimal equivalen a 5 y D. Así, al programar en código de lenguaje máquina, 01011101 es 5D. Éste constituye el procedimiento más lento de desarrollar programas, pero probablemente sea la manera de obtener los tiempos de ejecución más rápidos. Lo que sigue corresponde a extractos tomados de programas típicos:

```
BASIC
100 INPUT "Entre las horas trabajadas"; HORAS
200 PAGA = HORAS * TASA
```

La primera línea visualiza en pantalla un mensaje que invita al usuario a dar entrada a las horas trabajadas. Acepta la entrada y luego, en la segunda línea, la multiplica por la tasa de la paga (introducida previamente) para dar la cifra global de la paga.

```
Lenguaje ensamblador
MVL C, 01
CALL 05
```

La primera de las instrucciones anteriores desplaza el valor "1" a una parte de la memoria del microprocesador denominada "registro C". La segunda instrucción entrega el control al sistema operativo. Entonces el sistema operativo vuelve a entregar el control a su programa. Lo que sigue es la traducción directa de las instrucciones en lenguaje ensamblador que citamos arriba:

```
Código de lenguaje máquina
0E01
CD0500
```

Una traducción en código de lenguaje máquina de las dos sentencias en BASIC comprendería muchas órdenes. Evidentemente, es preferible que este trabajo lo realice un compilador o un interpretador, en lugar de escribirlo en lenguaje ensamblador o en código de lenguaje máquina.

Algunas órdenes del sistema operativo son tan enigmáticas como los programas en lenguaje ensamblador. He aquí algunos ejemplos tomados del CP/M (*Control Program for Microcomputers;* programa de control para microordenadores):

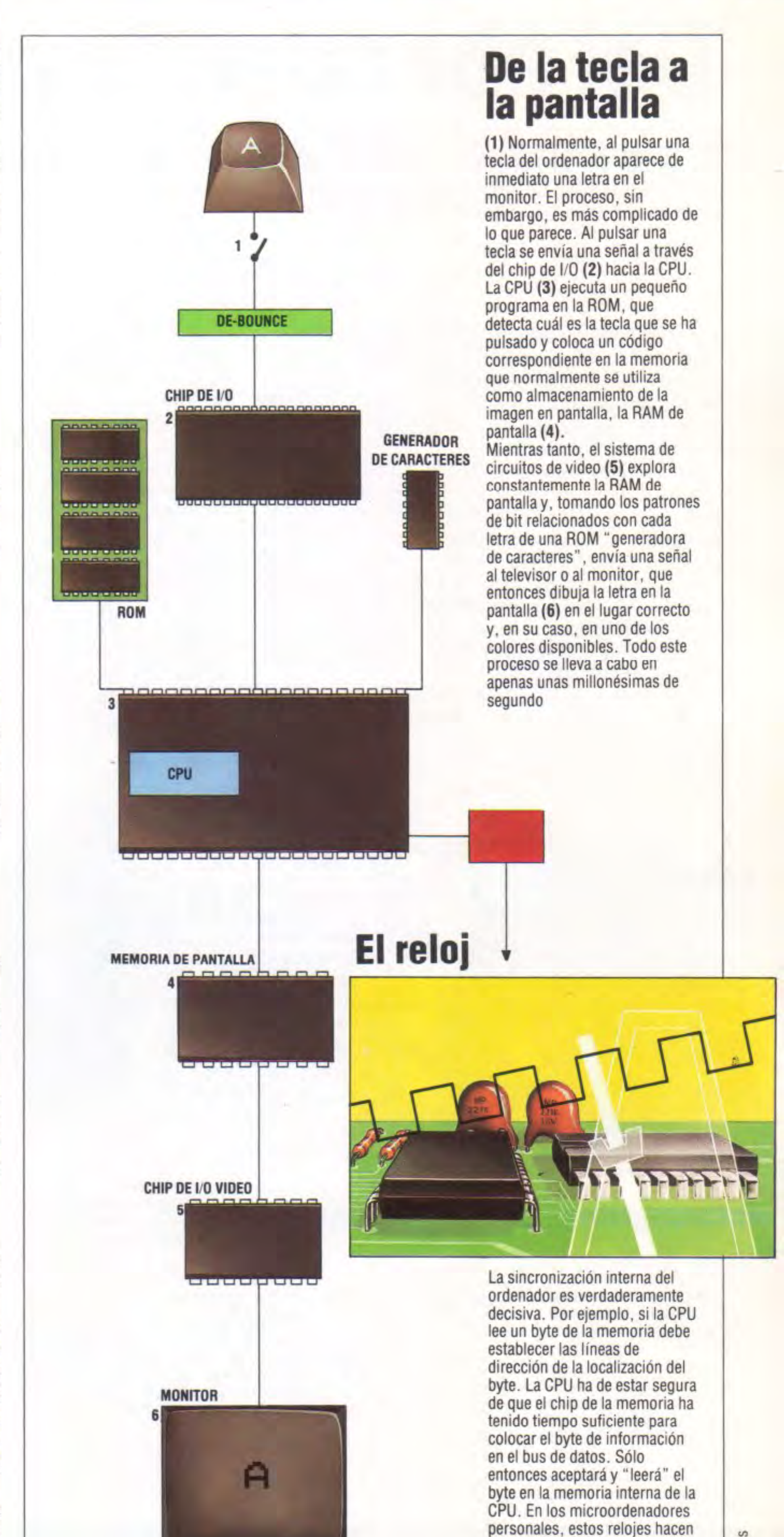
```
DIR * . BAS
```

Esto significa "liste todos los ficheros de la unidad de disco cuyo sufijo sea BAS". DIR es la abreviatura de "directorio".

A pesar de esta forma más bien extraña de elaborar las órdenes, el CP/M está instalado en gran número de máquinas. Ofrece enormes ventajas para los escritores de software profesionales. Los programas se pueden transportar fácilmente de una máquina a otra, con la única condición de que estén escritos de manera tal que entreguen al CP/M el control de la manipulación de los discos, el teclado, la impresora y la pantalla.

Los fabricantes que presenten una máquina nueva pueden beneficiarse del software de base ya existente, y los interesados pueden adquirir su ordenador en la confianza de que con toda seguridad podrán satisfacer sus necesidades en cuanto a software.

De la tecla a la pantalla



(1) Normalmente, al pulsar una tecla del ordenador aparece de inmediato una letra en el monitor. El proceso, sin embargo, es más complicado de lo que parece. Al pulsar una tecla se envía una señal a través del chip de I/O **(2)** hacia la CPU. La CPU **(3)** ejecuta un pequeño programa en la ROM, que detecta cuál es la tecla que se ha pulsado y coloca un código correspondiente en la memoria que normalmente se utiliza como almacenamiento de la imagen en pantalla, la RAM de pantalla **(4)**.
Mientras tanto, el sistema de circuitos de video **(5)** explora constantemente la RAM de pantalla y, tomando los patrones de bit relacionados con cada letra de una ROM "generadora de caracteres", envía una señal al televisor o al monitor, que entonces dibuja la letra en la pantalla **(6)** en el lugar correcto y, en su caso, en uno de los colores disponibles. Todo este proceso se lleva a cabo en apenas unas millonésimas de segundo

El reloj

La sincronización interna del ordenador es verdaderamente decisiva. Por ejemplo, si la CPU lee un byte de la memoria debe establecer las líneas de dirección de la localización del byte. La CPU ha de estar segura de que el chip de la memoria ha tenido tiempo suficiente para colocar el byte de información en el bus de datos. Sólo entonces aceptará y "leerá" el byte en la memoria interna de la CPU. En los microordenadores personales, estos relojes hacen tictac cuatro millones de veces por segundo

Steve Cross

Del ábaco al Apple

El microchip moderno debe su existencia a la genialidad de los inventores, cuyo trabajo abarca tres siglos

Cuando se está enamorado se piensa que puede conseguirse todo; al menos eso es lo que creía el matemático inglés Charles Babbage en 1830. La consecuencia de ello fue que estuvo a punto de construir el primer ordenador del mundo con cien años de adelanto en relación con su época.

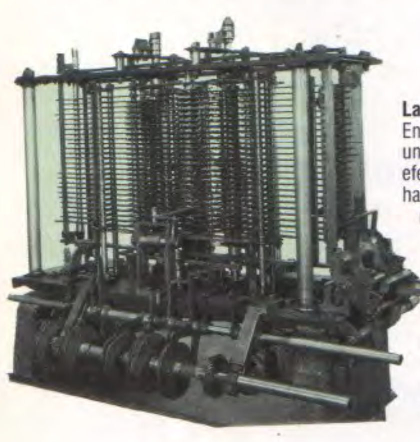
El diseño de Babbage tenía varios defectos: uno de ellos residía en el hecho de que el ordenador debía ser mecánico; otro era consecuencia de las limitaciones de la ingeniería de aquellos tiempos. Pero, a pesar de estos problemas, Charles Babbage construyó una máquina que impresionó tanto al gobierno británico de aquella época que enseguida le concedió una subvención de 1 500 libras esterlinas, que posteriormente aumentó a 17 000. (En la actualidad esta suma equivaldría a 1 700 000 libras.)

Sin embargo, la historia de la informática se remonta a mucho tiempo antes. Un ordenador es una máquina a la que se le puede decir cómo calcular una serie de números, capaz de recordar el cálculo que ha realizado y que se puede adaptar para calcular otra serie de números. El ejemplo más remoto es el ábaco, que ya se utilizaba en el año 2000 a. C. y que todavía se usa en Japón y en Europa oriental. El ábaco es una clase de ordenador muy útil, porque uno puede ver físicamente la suma en los alambres: la posición de las bolas movibles constituye una "memoria" de la suma. Pero el ábaco no es automático y no es útil para manejar cifras elevadas.

El matemático y filósofo francés Blaise Pascal inventó en 1642 la primera calculadora mecánica del mundo, posiblemente para complacer a su padre, que era el inspector de hacienda de la zona. La máquina funcionaba a la perfección: transportaba los números de la columna de las unidades hasta la columna de las decenas mediante un mecanismo de trinquete, más o menos de la misma forma en que transporta los números el velocímetro de un automóvil, y era totalmente funcional. Blaise la denominó *pascalina*.

A pesar de que la pascalina no se vendió bien, despertó un gran interés a nivel científico y durante unos años se introdujeron muchas mejoras en la calculadora original. Sin embargo, no se produjo ningún ade-

Blaise Pascal
La "pascalina" fue la primera calculadora mecánica del mundo y fue diseñada en 1642 por el francés Blaise Pascal. Su deseo original era crear una máquina de dividir y multiplicar, además de sumar y restar. La pascalina se valía de un punzón para hacer mover las ruedas y tenía un mecanismo especial que llevaba los dígitos de una columna a la otra. El rey de Francia le concedió a Pascal una patente para que pudiera comercializar su calculadora, pero en el aspecto financiero la máquina nunca fue un éxito



La máquina analítica
En 1834 Charles Babbage diseñó una máquina analítica que podía efectuar cálculos con números de hasta 80 dígitos. Incorporaba muchas de las características del ordenador moderno. Los "programas" se controlaban mediante tarjetas perforadas y los resultados se imprimían automáticamente. También tenía un "molino" aritmético y dispositivos de almacenamiento separados



Cortesía Science Museum

Cronología de la informática

1000 A. C.
EL ÁBACO
Aún en uso hoy en día, calcula y almacena números mediante bolas movibles

1614
LOGARITMOS
John Napier los inventa como medio para simplificar la división en resta y la multiplicación en suma

1623
NÚMEROS BINARIOS
Francis Bacon utiliza por primera vez la aritmética con base 2

1642
MÁQUINA DE SUMAR
Blaise Pascal inventa la "pascalina", primera calculadora mecánica

1671
CALCULADORA
Gottfried W. Leibniz logra construir una máquina que multiplica y divide

1802
TARJETAS PERFORADAS
Joseph Jacquard construye un telar que almacena los patrones de tejido en tarjetas perforadas

1822
MÁQUINA DIFERENCIAL
Charles Babbage produce su primera calculadora matemática

1000 A.C. // 1614 1623 1642 1671 1802 1822

lanto trascendente hasta que Charles Babbage y Ada Lovelace comenzaron a pensar en el asunto.

Charles Babbage nació en 1791 en el seno de una familia adinerada. Dispuso de toda clase de facilidades para acceder a una completa educación, y muy pronto demostró ser un genio de las matemáticas; quizá por la frustración que le producía tener que corregir los muchos errores que hallaba en las tablas logarítmicas, se concentró en la construcción de una máquina que pudiera hacer menos monótona la faena de calcular.

En 1822 presentó en la Royal Astronomical Society su primer modelo de un ingenio "diferencial", una máquina que efectuaba los cálculos necesarios para construir tablas logarítmicas. El nombre proviene de una técnica matemática abstracta conocida como método de las diferencias. La Royal Astronomical Society lo alentó para que creara máquinas nuevas y más perfeccionadas.

Trabajando en equipo con Ada Lovelace, hija de Lord Byron, se embarcó en un proyecto aún más ambicioso: construir un ingenio "analítico". Esta máquina se diseñó para calcular los valores de funciones matemáticas mucho más complejas que las logarítmicas.

La creación de esta máquina estuvo plagada de problemas desde el principio. Simplemente, no funcionaba. Al observar los bocetos del ingenio que se han conservado, se deduce que era enorme y que ocupaba todo el inmenso taller que Babbage había construido en su domicilio. Era preciso tornear de forma especial centenares de rodillos, varillas y engranajes, y la tecnología metalúrgica de aquella época no había alcanzado todavía los niveles necesarios de desarrollo para realizarlo satisfactoriamente. Cuando construyó su maqueta a escala, Babbage pudo soslayar las pequeñas inexactitudes de la máquina, pero, una vez construida, estos mínimos desajustes demostraron ser importantes.

No obstante, Babbage se hallaba en el buen camino y, si hubiera conseguido que todos los componentes del ingenio armonizaran bien, probablemente su máquina analítica hubiera funcionado. Gran parte de la arquitectura lógica y la estructura de diseño de los ordenadores actuales se remonta al trabajo de Charles Babbage, a quien se considera como uno de los padres de la informática moderna.

Uno de los presentimientos que alentaron a Babbage durante su trabajo fue la idea de que a su ingenio se lo podría "programar" o "enseñar" a efectuar cualquier tarea matemática. De haber sido capaz de demostrar esto, o si hubiera podido construir una máquina que efectivamente lo hiciera, los victorianos habrían gobernado su imperio mediante un ordenador a vapor.

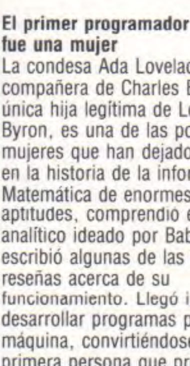
No fue hasta 1936 que se confirmó el presentimiento de Babbage. La prueba de su viabilidad apareció en *On Computable Numbers*, modesta publicación editada por un joven matemático de Cambridge, Alan Turing. El nombre de Turing es prácticamente desconocido para el gran público, pero su contribución fue fundamental para el desarrollo de las ideas cuya concreción hizo que el ordenador se convirtiera en realidad. Durante mucho tiempo los científicos habían pensado que las matemáticas no eran un arte misterioso sino una ciencia exacta regida por reglas lógicas, y que si se le proporcionaba a una máquina esas reglas y un problema, ésta debería ser capaz de resolverlo. Sin embargo, los esfuerzos de los matemáticos más capaci-

Alan Turing y el Colossus
Alan Turing demostró que una serie de instrucciones sencillas podían resolver cualquier problema complejo. Él y su equipo desarrollaron el Colossus, uno de los primeros ordenadores, que en la fotografía se ve en pleno funcionamiento durante la segunda guerra mundial. Esta enorme máquina empleaba 1 500 válvulas, que se quemaban a razón de una cada pocos minutos. El Colossus podía procesar 5 000 caracteres por segundo y tenía la misión de descifrar el código alemán Enigma

Cortesía Science Museum



Cortesía Colección HMG



El primer programador fue una mujer
La condesa Ada Lovelace, compañera de Charles Babbage y única hija legítima de Lord Byron, es una de las pocas mujeres que han dejado su huella en la historia de la informática. Matemática de enormes aptitudes, comprendió el ingenio analítico ideado por Babbage y escribió algunas de las mejores reseñas acerca de su funcionamiento. Llegó incluso a desarrollar programas para la máquina, convirtiéndose así en la primera persona que programó ordenadores



1834
MAQUINA ANALÍTICA
Babbage formula muchas de las ideas sobre las que se basa el ordenador moderno

1835
PRIMER PROGRAMA
Ada Lovelace escribe el primer programa

1847
EL ÁLGEBRA BOOLEANA
George Boole inventa una forma de matemáticas que favorece el estudio del cálculo

1890
PROCESAMIENTO DE LA INFORMACIÓN
Hermann Hollerith utiliza tarjetas perforadas para llevar a cabo el censo de Estados Unidos

1900
LA MEMORIA MAGNÉTICA
Valdemar Poulsen realiza las primeras grabaciones magnéticas de información

1906
LA VÁLVULA
Lee de Forest inventa el interruptor electrónico

1931
EL ANALIZADOR DIFERENCIAL
Vannevar Bush inventa el analizador diferencial, para resolver una familia de ecuaciones diferenciales

1936
LA MÁQUINA DE TURING
El matemático inglés Alan Turing establece los principios teóricos del ordenador

1938
CIRCUITOS ELECTRÓNICOS
Claude Shannon demuestra que los circuitos electrónicos de conmutación pueden realizar operaciones lógicas

1941
CONRAD ZUSE
Construye el primer ordenador electrónico programable

1834 1835 1847 1890 1900 1906 1931 1936 1938 1941

tados por desarrollar una máquina de esas características habían resultado vanos. Turing decidió enfocar el problema desde una perspectiva diferente. Estudió la clase de problemas que podría resolver una máquina siguiendo reglas lógicas e intentó hacer una lista de todas ellas. Si éstas comprendían la totalidad de las matemáticas, la presunción podría confirmarse.

Turing dirigió un equipo de investigación en Buckinghamshire y desarrolló el invento más secreto de la segunda guerra mundial, el Colossus, primer ordenador electromecánico del mundo. Fue esta máquina la que descifró los mensajes alemanes en código Enigma durante el conflicto bélico.

Acabada la guerra, Turing se estableció en Estados Unidos para poner su genio al servicio del primer proyecto norteamericano de informática. Con su ayuda se construyó el primer ordenador norteamericano. Éste se llamó ENIAC y se desarrolló en la Universidad de Pennsylvania. Empleaba 18 000 válvulas, ¡cada una de las cuales se quemaba en dos minutos!

Una de las razones por las cuales el nombre de Turing es virtualmente desconocido, es que trabajaba para la MI6 y tanto su trabajo como él mismo estaban rodeados por el mayor secreto. El gobierno británico no dio a conocer hasta 1975 detalles de la labor pionera de Turing en el campo de la informática.

La informática se perfeccionaba progresivamente, pero no fue hasta 1947, cuando se inventó el transistor de silicio, que el cálculo rápido se hizo realidad.

Los transistores cumplen la misma función de las válvulas, pero a una velocidad superior, con mayor fiabilidad y sin generar calor. Al igual que las válvulas, son interruptores electrónicos que pueden encenderse y apagarse y que se pueden utilizar para representar los ceros y los unos del código binario. Durante los años cincuenta y primeros sesenta se diseñaron ordenadores cada vez más rápidos y de mayores dimensiones, que eran utilizados tanto por las grandes empresas como por los gobiernos.

Mediada la década de los sesenta, los científicos comprendieron que un circuito electrónico miniaturizado trabajaría con similar eficacia. Respaldados por los miles de millones de dólares invertidos en la carrera espacial, los laboratorios comenzaron a experimentar colocando diseños de circuitos en un único chip de silicio y grabando luego en él el diseño. Antes de que finalizara la década había nacido este "circuito integrado" y la informática experimentó un impresionante salto adelante. El desarrollo de un "circuito dentro de un chip" llevó, naturalmente, a un "circuito múltiple dentro de un chip", y la inevitable consecuencia de disponer varios chips juntos por capas condujo a la invención del microprocesador.

Aunque la tecnología de los microchips guarda poco parecido con el gigantesco ingenio analítico que construyeron Babbage y Ada Lovelace, y apenas si se parece un poco más al Colossus de Turing, la "arquitectura" práctica que creó Babbage aún se utiliza en el microprocesador actual. La teoría que la hizo posible, la comprobación matemática de Turing de la viabilidad de la informática, aún no ha sido superada.

El "arquitecto" del ordenador moderno
A John von Neumann se le invitó a colaborar en el proyecto ENIAC como asesor en los problemas del diseño lógico. Su informe fue decisivo en el posterior desarrollo del ordenador. Von Neumann aconsejó que los ordenadores futuros almacenaran internamente sus programas y que tanto la información como los programas se representaran mediante números binarios

Cortesía de la London Mathematical Society



Steve Wozniac
En la industria de los ordenadores, a Steve Wozniac con frecuencia se le considera una leyenda viviente. En la fotografía aparece junto a su primer invento, el Apple I (hoy una pieza de coleccionista). Wozniac diseñaba ordenadores en la escuela. Aunque nunca estudió ingeniería, fue él quien se percató de que, para que el ordenador se convirtiera en un aparato personal y de precio asequible, era necesario miniaturizarlo. El Apple II, que desarrolló en 1976, dispuso de la primera unidad de disco para ordenador personal y actualmente sigue siendo uno de los modelos más vendidos en todo el mundo

El Sinclair Spectrum
Desarrollado sólo 40 años después de aquellos incómodos primeros ordenadores que ocupaban habitaciones enteras, el Spectrum es pequeño, compacto y económico, y se constituyó en el primer ordenador personal en color puesto a la venta a un precio módico. Los primeros ordenadores eran máquinas gigantescas y complicadas, construidas por los gobiernos e instituciones del estado, y



funcionaban a puerta cerrada. En la actualidad, gracias a máquinas como el Spectrum, es posible que muchas familias puedan comprar un ordenador personal

1943 COLOSSUS
Primer ordenador electromecánico del mundo

1946 ENIAC
Primer ordenador a válvulas (Estados Unidos)

1947 VON NEUMANN
Su informe sobre el ENIAC ejerce gran influencia en el diseño y desarrollo de los ordenadores

1948 EL TRANSISTOR
Lo inventa William Shockley

1949 MARK I: UNIVERSIDAD DE MANCHESTER
Se construye el primer ordenador que responde al diseño de Von Neumann

1951 ACE
Uno de los primeros ordenadores diseñados para cálculo veloz

1957 FORTRAN
El primer lenguaje para ordenador de alto nivel

1957 IBM
Crea su primer ordenador

1962 DISCOS MAGNÉTICOS
Se usan por primera vez con el ordenador ATLAS

1963 CIRCUITOS INTEGRADOS
Se fabrican desde 1963

1964 BASIC
Se inventa el lenguaje para micros más popular

1972 LSI
Large Scale Integration (circuito de integración a gran escala): llega el chip

1982 SPECTRUM
Primer microordenador popular en color

1990 QUINTA GENERACIÓN
Los ordenadores se programan a sí mismos

1943 1946 1947 1948 1949 1951 1957 1962 1963 1964 1972 1982 1990



BBC Modelo B

Gracias a su excelente configuración técnica, avalada por los recursos de comercialización y programación de la BBC de Londres, este microordenador se ha hecho muy popular

El BBC Modelo B, recientemente introducido en el mercado español, lo fabrica Acorn Computers en Cambridge (Inglaterra) y lo comercializa la BBC. Originalmente había dos modelos, pero el Modelo A, más barato y menos sofisticado, está quedando desfasado.

En Gran Bretaña, el Modelo B se ha vendido muy bien en las escuelas y figura en la lista de ordenadores aptos para esa finalidad aprobada oficialmente por el gobierno. Para esta máquina se ha desarrollado una cantidad importante de software educativo, desde lenguajes de programación hasta paquetes de programas para "aprendizaje mediante ordenador".

Las especificaciones técnicas del Modelo B se consideran excelentes. En particular, el lenguaje de programación BASIC del BBC está muy bien equipado con órdenes para afrontar las funciones especiales. También simplifica la tarea de desarrollar y editar programas.

Existen ocho modalidades diferentes para gráficos. Esto significa que el usuario puede elegir entre una resolución alta, media o baja, aunque en el primer caso el número de colores disponibles es limitado. La máxima resolución que se puede obtener es de 640 × 256. La mayoría de programadores optan por un televisor normal, pero para obtener los mejores resultados en los gráficos del Modelo B es recomendable usar un monitor exclusivo. Existen órdenes para trazar en la pantalla líneas y círculos y crear una gran variedad de imágenes.

Su fuente de alimentación eléctrica va alojada en el interior de la carcasa, lo que le proporciona una apariencia bien proporcionada y autosuficiente; pero las interfaces situadas detrás y debajo de la carcasa son más numerosas que en otras máquinas. Ello significa que existe una gran cantidad de dispositivos idóneos para ampliar el ordenador estándar, incluyendo diversas versiones de unidad de disco, aparte de la que la BBC fabrica especialmente.

Además de las interfaces para unidad de disco, impresora y un dispositivo análogo, como un equipo de laboratorio o un instrumento de medición, existe una configuración para trabajar en red. Ésta es ideal para trabajar en las escuelas, puesto que varios usuarios pueden compartir la misma impresora o la misma unidad de disco.

Por último, existe "el tubo", una sofisticada interface para conectar un microprocesador alternativo, ya sea para conseguir una mayor velocidad de cálculo o bien para ejecutar el software escrito para otras máquinas. No obstante, al parecer son muy pocos los programadores que aprovechan este dispositivo.



El teclado del BBC Modelo B
El teclado es uno de los puntos fuertes del BBC Modelo B, en términos de trazado, facilidades y calidad de construcción. Las teclas están moldeadas convenientemente, gracias a lo cual hasta una persona que esté acostumbrada a mecanografiar al tacto se sentirá muy cómoda. Las cuatro teclas con flechas sobre el lado derecho son para hacer mover el cursor por la pantalla cuando se editan textos o programas.
La hilera superior, de diez teclas rojas de función programable, es particularmente útil para programas educativos, dado que el usuario debe tan sólo escoger la respuesta adecuada entre 10 posibilidades.
Una característica muy agradable es la inclusión de tres LED (Light Emitting Diodes; diodos emisores de luz) para indicar si el motor de la cassette está funcionando y si se han activado las teclas de retención de mayúsculas





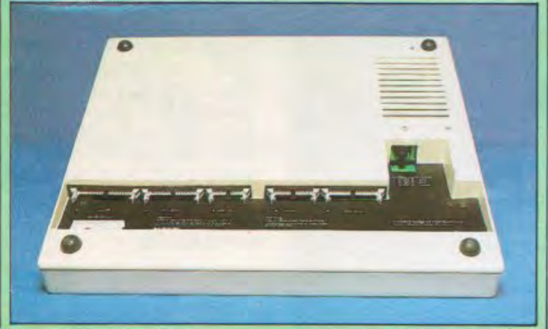
La unidad de disco
La unidad de disco del micro BBC es un accesorio muy atractivo pero caro, y sólo puede almacenar 100 Kbytes. Sin embargo, existe la posibilidad de utilizar las unidades de disco más baratas que proporcionan otros fabricantes

Entrada analógica
Permite que el ordenador lea el voltaje de un componente no digital del equipo, como por ejemplo de un dispositivo sensor de calor. Se utiliza básicamente en los laboratorios y en trabajos de experimentación

Controlador video
Este chip recibe la información de la memoria del usuario y la convierte en una señal de video para que se pueda visualizar

Chips importados
Si observa atentamente los chips, verá que la fabricación de microprocesadores es muy internacional. El BBC Modelo B contiene chips fabricados en Malaysia, Japón, Portugal, Escocia y Estados Unidos

Chips interfaces
Adaptadores versátiles para interfaces, como este MOS 6522, vigilan la conexión por interface con dispositivos externos. Si bien no son procesadores, estos chips son tan sofisticados como el propio microprocesador



Conexiones para periféricos
Vista de la cara inferior del BBC Modelo B, en la que se ven los enchufes en donde se pueden conectar los periféricos

Ian McKinnell

Conexión impresora
Aquí se puede enchufar cualquier impresora de señal en paralelo

Conexión dispositivos
Para la experimentación con dispositivos digitales y circuitos lógicos caseros

BBC Modelo B

DIMENSIONES
409 × 358 × 78 mm

PESO
3,70 kg

CPU
6502A

VELOCIDAD DEL RELOJ
2 MHz

MEMORIA
32 Kbytes de RAM, 32 Kbytes de ROM, incluyendo BASIC y un sofisticado sistema operativo

VISUALIZACION EN VIDEO
Ocho modalidades diferentes para gráficos, que proporcionan una gran variedad de visualizaciones. Mayor superficie de textos: 32 líneas de 80 caracteres. Resolución más alta para gráficos: 640 × 256 pixels. Hasta 16 colores con una resolución más baja

INTERFACES
Televisor, monitores monocromáticos y en color, unidad de disco, impresora, entrada analógica, conexión dispositivos, tubo (para conectar microprocesadores adicionales)

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
LISP, FORTH, LOGO

VIENE CON
Cables para cassette y televisor. Guía para el usuario. Folleto y cassette de demostración

TECLADO
Estilo máquina de escribir, con 74 teclas móviles, incluyendo 10 teclas de función programable

DOCUMENTACION
La Guía para el Usuario del microordenador BBC parece haber sido escrita por personas de gran experiencia y que daban por sentado que todos sus lectores ya eran profesionales de la informática.
Varios capítulos se dedican a la utilización especializada de los programas del sistema, que controlan las sofisticadas configuraciones de la máquina para gráficos, sonidos y entrada/salida.
Se incluye una explicación detallada y muy completa de la forma en que opera y está programado el microprocesador 6502 y, a diferencia de muchas de las secciones de este tipo de otros manuales, no se trata tan sólo de una copia de la documentación Rockwell original

Puertas y sumadores

Los números binarios, compuestos de ceros y unos, se pueden sumar entre sí mediante la sencilla lógica del AND, OR y NOT

En un artículo anterior (véase p. 68) hemos visto cómo los relativamente sencillos circuitos de transistor se pueden utilizar para tomar decisiones lógicas como AND, OR y NOT. Lo sorprendente es que estas mismas "puertas lógicas" también constituyen los bloques de construcción que se emplean para realizar las funciones aritméticas en el interior del ordenador. Lógicamente, las entradas de las puertas son, bien de voltaje cero, para representar "falso", o bien de voltaje positivo, para representar "verdadero". La ausencia de voltaje se suele simbolizar con un cero (0) y el voltaje positivo con un uno (1). Cuando las puertas lógicas se utilizan para la aritmética, se emplean los mismos ceros y unos, pero entonces representan literalmente los unos y los ceros que se suman.

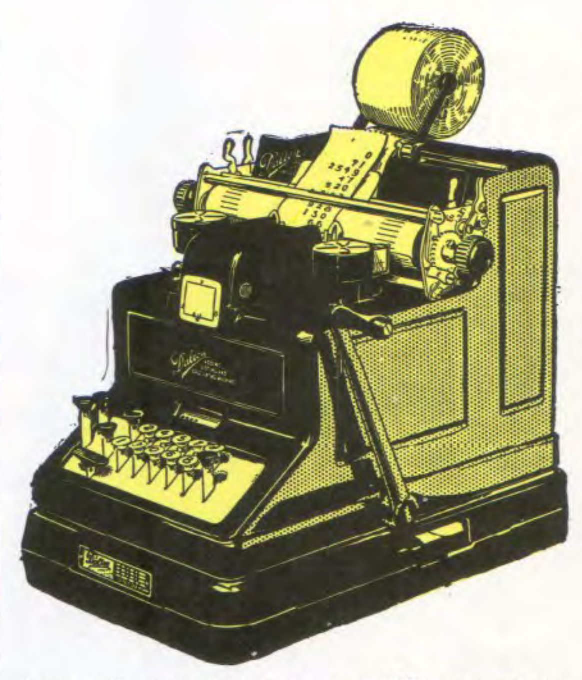
Si deseamos sumar dos dígitos binarios, sólo habrá dos entradas en el circuito de suma, y únicamente podrá haber cuatro combinaciones de entrada: 0 + 0, 0 + 1, 1 + 0 y 1 + 1. Estudiando la aritmética binaria hemos aprendido que 0 más 0 es igual a 0 (como en aritmética decimal). Sabemos también que 0 más 1 (o 1 más 0) es igual a 1 (igual que en aritmética decimal). La diferencia, respecto a la aritmética que hemos aprendido en la escuela, es que en binario 1 más 1 es igual a 0 y llevamos 1. La demostración aritmética de estas cuatro sumas sería la siguiente:

X		Y		Z
0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

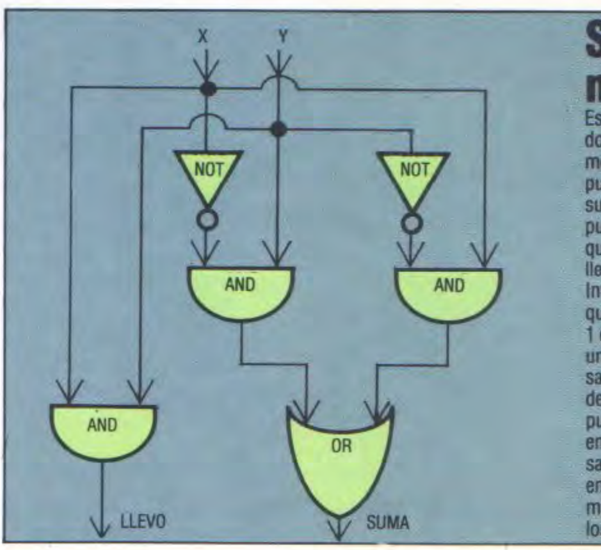
Si hubiéramos de utilizar una puerta OR para efectuar la suma, obtendríamos una salida falsa (0) si ambas entradas fueran falsas (0), y una salida verdadera (1) si alguna de las entradas fuera verdadera (0 + 1 o 1 + 0).

De momento, para sumar dos dígitos binarios parecería perfectamente adecuado utilizar una puerta OR simple. Pero veamos. Si ambas entradas fueran verdaderas, la salida de una puerta OR simple también sería verdadera, pero en aritmética binaria esa respuesta estaría equivocada. La respuesta correcta sería 0 y llevo 1. Una puerta OR simple daría una respuesta correcta para tres de las cuatro combinaciones de entrada posibles, pero tres de cuatro no es un nivel satisfactorio.

Lo que se necesita es un circuito que dé una respuesta de 0 si ambas entradas son 0, y una respuesta de 1 si una de las entradas es 0 y la otra es 1, y una respuesta de 0 si ambas entradas son 1 (como en la tabla de verdad anterior). Esto no es tan difícil como parece. Si tenemos dos puertas AND, con las dos entradas yendo a ambas puertas, pero invirtiendo una de las entradas a través de una puerta NOT por una de las puertas AND, y si la otra entrada se invierte a través de otra puerta NOT que va hacia la otra puerta AND (véase la ilustración), obtendríamos una situación en la cual un 0 en ambas entradas dará una salida falsa en las dos puertas AND, y, del mismo modo, un 1 en ambas entradas dará una salida falsa para ambas puertas. Por otra parte, un 0 en una entrada y un 1 en la otra darán dos entradas verdaderas para una de las puertas AND. Una de estas puertas producirá, por tanto, una salida verdadera. Si las dos puertas AND tuviesen conectadas sus salidas con una puerta OR, la salida de la puerta OR sólo sería verdadera si una, y sólo una, de las dos entradas fuera verdadera.



La calculadora de mesa
Hasta que hace poco tiempo se inventara la calculadora electrónica, la máquina de sumar mecánica (o caja registradora) era un objeto muy común en las tiendas y oficinas. Exceptuando unos pocos refinamientos, esta máquina no sufrió modificaciones esenciales durante trescientos años, funcionando mediante una serie de engranajes y ruedecillas. Muy pronto se advirtió el potencial comercial de la calculadora. Pascal inventó la primera máquina de sumar para que su padre, que era inspector de hacienda, la utilizara en su despacho. Cuando Leibniz desarrolló la multiplicación y la división, la calculadora se introdujo en el mundo empresarial



Sumador medio
Es un dispositivo para sumar dos números binarios mediante una combinación de puertas lógicas. Se llama sumador medio porque no puede arrastrar los dígitos que a menudo hay que llevarse al sumar números. Intente sumar 1 y 1. Recuerde que una puerta NOT invierte el 1 en 0 y el 0 en 1. Para que una puerta AND produzca una salida 1, las dos entradas han de ser 1. La salida de una puerta OR será 1 si una entrada o ambas son 1. La salida sólo será 0 si ambas entradas son 0. La ilustración muestra el camino que siguen los dígitos

Mark Watkinson

Grabando

El gran temor del programador es perder la obra de arte que ha creado en la pantalla. Las cassettes ofrecen la solución

Altavoz
En la mayoría de las grabadoras de cassette el altavoz queda desconectado cuando se acopla el aparato a un ordenador o a un equipo de alta fidelidad

Cabeza de borrado
Estando en modalidad de grabación, elimina de la cinta cualquier señal grabada previamente

Contador de vueltas
Un dispositivo esencial si se piensa almacenar diversos programas en una misma cinta

Cabeza de grabación/reproducción
Esta cabeza de doble función graba las señales sonoras en la cinta magnética y las reproduce

La Hobbit
La Hobbit de la Ikon es una grabadora de cassette exclusiva, es decir, diseñada exclusivamente para almacenar programas de ordenador. Es superior a la grabadora de cassette normal, porque la Hobbit está totalmente controlada por un software. No es necesario pulsar "wind", "rewind", "play" ni "record", puesto que la Hobbit realiza todas estas funciones por sí misma. Si desea cargar (LOAD) un programa, deberá digitar el nombre del mismo y la Hobbit buscará en su propio catálogo hasta localizarlo. Luego rastrea la cinta hasta colocarla en la posición correcta

Eje de arrastre
Eje de gran precisión que rota a una velocidad cuidadosamente controlada para hacer pasar por la cabeza de grabación/reproducción 1 ⅞ de pulgada de cinta por segundo

Volumen
Regula el volumen de reproducción y ha de determinarse con sumo cuidado, ya que de lo contrario se corre el riesgo de que las cassettes no se carguen correctamente

Ian McKinnell

Cuando se digita un programa en un ordenador personal, ya sea tomándolo de un libro, de una revista o confeccionándolo uno mismo, la información se almacena en la RAM. Mientras el ordenador está encendido, el programa está a salvo; pero desaparece en el momento en que usted desconecta la máquina. Esto puede suponerle un gran contratiempo: ¡toda una sesión de programación perdida para siempre! La próxima vez que desee utilizar ese programa habrá de digitarlo todo de nuevo. Para evitar este problema, los fabricantes de ordenadores personales suelen incorporar un procedimiento en virtud del cual se puede almacenar con carácter más permanente el contenido de la memoria del ordenador.

Motor
El motor acciona el eje de arrastre a una velocidad constante y también hace girar los tambores para bobinar y rebobinar la cinta

El procedimiento más común de almacenamiento es el de la cinta de cassette. Escogido originalmente por su gran disponibilidad y su bajo precio, en la actualidad casi todos los ordenadores personales que se venden incorporan este sistema. La forma en que cada ordenador almacena su información varía ligeramente; por ejemplo, un programa creado y almacenado en un ordenador Commodore no se puede cargar en un Spectrum. Sin embargo, el procedimiento que se utiliza para convertir el programa en una forma almacenable, es casi universal.

La clase de grabadoras de cassette que utilizan la mayoría de los ordenadores personales es, obviamente, más adecuada para almacenar sonidos; sin embargo, en el interior del ordenador el programa se almacena en forma de números binarios. Éstos han de convertirse en sonidos para que el ordenador pueda reconocer la diferencia entre un bit que "se enciende" y un bit que "se apaga": los ceros y los unos binarios. Para hacerlo, el procedimiento más sencillo consiste en crear un sonido que represente un 1 y otro que represente un 0.

David Weeks

Control remoto
Una configuración muy útil que permite que el ordenador personal controle a la grabadora de cassette

Conexión auricular
Muchos micros emplean esta salida para volver a alimentar los tonos al ordenador

Conexión micrófono
Utilizada a menudo para la entrada de datos del ordenador en la grabadora. No obstante, sólo debe emplearse si no se dispone de conexiones para DIN y auxiliares. Cuando se utiliza esta entrada es necesario realizar un cuidadoso ajuste de los mandos de tono y volumen

Por lo general, estos sonidos se escogen de manera que sean un tono de 2 400 ciclos para el 1 y un tono de 1 200 ciclos para el 0.

Cuando se digita en el ordenador la orden SAVE, lo primero que se grabará en la cinta son algunos segundos de tono constante. Esto se hace para que al reproducirle la cinta al ordenador en algún otro momento éste pueda percibir la diferencia entre la cinta virgen y la sección que aloja el programa. La primera información real que se graba es la serie de tonos que representan a los caracteres del nombre que le hemos dado al programa. Cada carácter está compuesto de un byte (ocho bits en total), de modo que para representar un carácter son necesarios ocho tonos. Sin embargo, para indicar el comienzo y el final de cada byte el ordenador normalmente coloca un tono extra a ambos extremos. Éstos se denominan *bits de principio y final* y su valor siempre es el mismo: ya 1 o 0, según de qué ordenador se trate.

El propio programa se almacena de forma bastante similar, excepto en ocasiones que se lo divide en segmentos. Normalmente estos segmentos son de 256 bytes cada uno y suelen incluir una información adicional que le permite al ordenador asegurarse de que está recargando la información correcta. El sistema empleado aquí es muy sencillo y se denomina *suma de comprobación*. El primer byte contiene el número de bytes de que consta el segmento, y el último byte contiene un número calculado especialmente que representa el total de todos los bytes sumados. Cuando el ordenador lee la cassette verifica si las cifras que figuran en la cinta coinciden con las que él mismo ha calculado y, en caso de que no coincidan, informa del error al usuario.

Algunos sistemas de cassette, como el del modelo BBC Micro, amplían esta comprobación hasta el extremo de otorgar un nombre y un número a cada segmento del programa. En el caso de que se produzca un error, no hay problema alguno, pues se puede rebobinar la cinta unas pulgadas y volver a intentarlo. En notorio contraste, otros sistemas de cassette ni siquiera muestran el nombre del programa que se está cargando.

La velocidad baudio

La velocidad a la cual se producen y se graban los tonos en la cinta se suele denominar, incorrectamente, *velocidad baudio*. El nombre proviene del código Baudot, que se utilizaba en las primeras versiones del telégrafo eléctrico, y que realmente se refiere al número de veces por segundo que cambia la señal. Una medida más exacta sería el número de bits que se graban por segundo. A mayor velocidad (entre 300 y 1 200 bits por segundo), más rápidamente se almacenarán los programas en cinta y menos tiempo se tardará en volver a cargarlos en el ordenador. Lamentablemente, la mayor velocidad de almacenamiento de los tonos incide en la fiabilidad; una velocidad de 1 200 bits por segundo es a la vez fiable y rápida. Algunos sistemas ofrecen dos velocidades, por lo general una velocidad lenta superfiable de 300 bits por segundo y una velocidad rápida de 1 200 o 2 400 bits por segundo. Para evitar imprevistos, los programas valiosos se pueden guardar en dos copias, una de cada tipo.

La cinta de cassette ha de ser de buena calidad; no hay ningún inconveniente en usar las cintas de sonido normales en vez de las cassettes especiales, siempre y cuando sean de una marca prestigiosa y su longitud no exceda de C-60.

La capacidad aproximada de una determinada longitud de cinta se establece dividiendo por 10 la velocidad de la interface para cassette. El resultado de esta división corresponderá a la cantidad de bytes que se pueden almacenar en la cinta por segundo; una C-60 de 30 minutos por cada lado, en la que la interface trabaje a 1 200 bits por segundo, podría retener 432 Kbytes de programa.

Almacenamiento seguro

El ordenador puede almacenar en su memoria miles de bytes de información y recordar dónde está localizado cada uno de ellos

La memoria del ordenador se puede describir en términos de almacenamiento a corto plazo y almacenamiento a largo plazo. La de largo plazo no pierde la información almacenada y la puede retener durante largos períodos, aun después de haberse apagado el ordenador. En esta categoría entran las cintas magnéticas y los discos flexibles.

Los ordenadores también necesitan una memoria rápida a corto plazo para el almacenamiento temporal de programas y resultados.

Otra forma de describir la memoria del ordenador consiste en considerarla en términos de memoria interna o memoria externa. La memoria interna está localizada dentro del ordenador y por lo general es totalmente "electrónica", mientras que la memoria externa es periférica o exterior al ordenador. La memoria externa por lo general es parcialmente mecánica, y comprende dispositivos tales como cassettes, unidades de disco flexible e incluso tarjetas de papel perforadas.

Generalmente la memoria electrónica interna se denomina memoria principal, mientras que la memoria externa se considera como una memoria secundaria o de apoyo. En la actualidad existen dos variedades principales de memoria interna: RAM y ROM.

EPROM
El problema de las memorias ROM normales es que poseen un contenido "incorporado", determinado en una etapa de la fabricación y que no se puede modificar. Las unidades EPROM (Erasable Programmable Read Only Memory: memoria de lectura sólo susceptible de borrar y programar) son mucho más flexibles. Una vez programadas, se pueden volver a programar borrando los contenidos y volviendo a "escribir" en ellas. Las unidades EPROM incorporan una "ventana" de sílice que permite que penetren los rayos ultravioletas, haciendo que se descarguen los condensadores que almacenan los bits en la EPROM. En ausencia de luz ultravioleta, los condensadores conservan su carga indefinidamente y se retienen los contenidos de la memoria

Tanto la RAM como la ROM son dispositivos totalmente electrónicos, fabricados en forma de chips de silicio y empaquetados en estuches plásticos rectangulares con cables paralelos de hojalata o chapados en plata. Existe mucha similitud en cuanto a la forma en que son seleccionados y "dirigidos" por la CPU del ordenador, pero de ello nos ocuparemos más adelante.

La principal diferencia funcional es que los chips de memoria ROM se utilizan para almacenar programas con carácter permanente. El patrón de unos y ceros de cada dirección de memoria se fija y se determina durante el proceso de fabricación y, en consecuencia, no se puede modificar. Las ROM son las "bibliotecas de referencia" del mundo de la informática. El ordenador puede consultar el contenido de la ROM, pero no puede "escribir" nada en ella.

ROM corresponde a las siglas de Read Only Memory (memoria de lectura solamente); la palabra lectura describe lo que hace el ordenador cuando "accede a" o recupera información de la memoria. Existen diversos tipos de ROM que difieren ligeramente entre sí; algunos de ellos permiten eliminar o borrar especialmente el programa interno y se pueden volver a programar. No obstante, una ROM bastante clásica es la 2364 de Intel. Este chip es una ROM de 65 536 bits, organizados en 8 Kbytes de 8 bits. Esto significa que los 64 Kbits están agrupados en bytes de 8 bits y que cada localización "dirigible" accede a o lee un byte entero. En matemáticas, $1\ K = 2^{10}$ (dos a la décima potencia) o 1 024, de modo que $64\ K = 64 \times 1\ 024$ o 65 536.

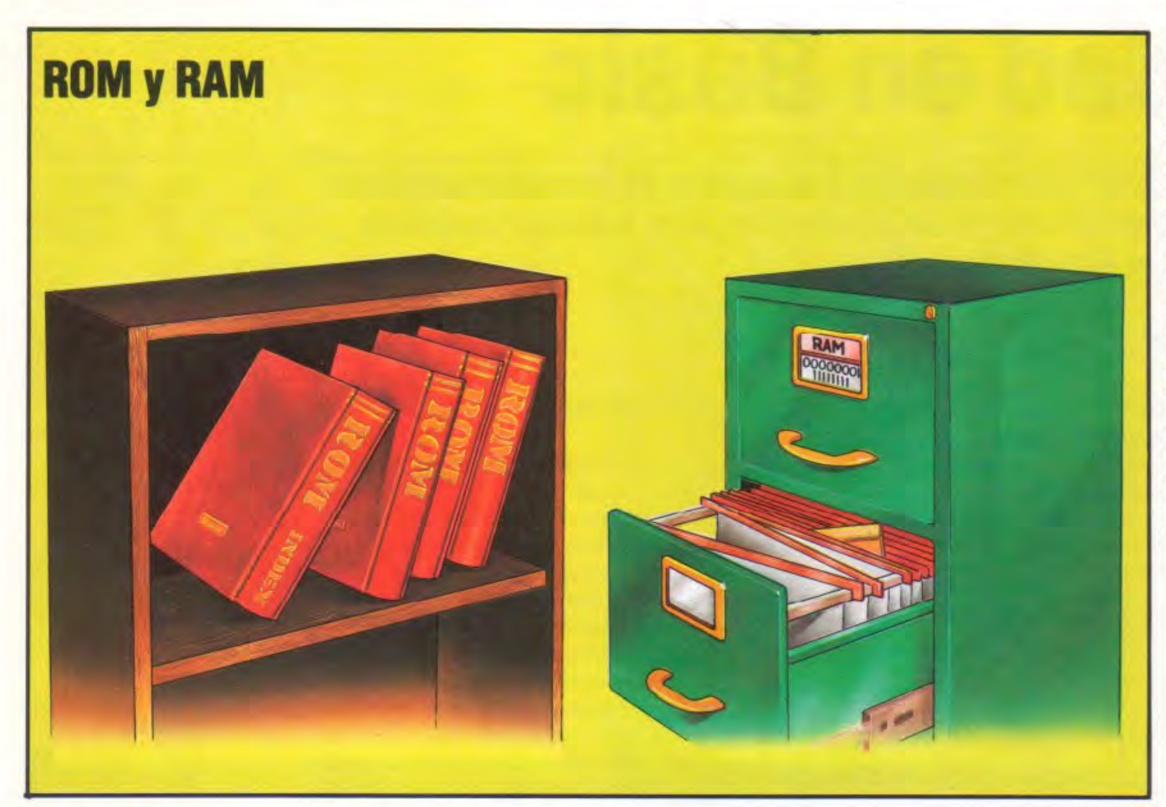
El ordenador, por lo tanto, ha de ser capaz de seleccionar cualquiera de las 8 192 (8 K) localizaciones de dirección. Una atenta lectura de las especificaciones del chip 2364 revela que posee 28 patillas, con una reservada para la fuente de alimentación eléctrica de +5 voltios y otra para la conexión a tierra. Esto deja un total de 26 patillas. Cada byte contiene ocho bits, de modo que cuando se lee un byte del chip, los ocho dígitos de ese byte han de traspasarse mediante cables desde el chip a la CPU. En consecuencia, hay ocho cables para trasladar los bits del byte que se le está leyendo a la CPU. Estos cables se denominan *bus de datos*. Ocho de las patillas del chip se dedican a esta función, una para cada uno de los bits del byte.

Esto deja un total de 18 patillas. Una patilla no es necesaria y no se conecta. Se conserva porque es más fácil fabricar chips con un número par de patillas. Cuatro patillas se utilizan para "seleccionar" el chip de diversas maneras. Éstas son la patilla de "habilitación de salida", la patilla de "habilitación del chip" y las dos patillas de "selección de chip". Estas patillas reciben las señales del ordenador para permitir que el chip sepa cuándo es requerido.

Las 13 patillas restantes son las patillas "de dirección". Cada patilla está conectada a un cable de "bus de direcciones", y éste transporta la dirección del byte requerido, codificada en forma binaria. Trece dígitos binarios pueden dar 2^{13} u 8 192 combinaciones exclusivas de unos y ceros, de modo que las 13 líneas de dirección alcanzan exactamente para seleccionar de forma exclusiva cada uno y la totalidad de los 8 192 bytes almacenados en la ROM.

Las unidades RAM son algo así como las pizarras del mundo de la informática. En ellas los programas y los datos se almacenan con carácter provisional, mientras el ordenador está en funcionamiento, y en ellas también "se escriben" temporalmente los resultados y

ROM y RAM



La memoria a largo plazo
La ROM (Read Only Memory: memoria de lectura solamente) se puede comparar con un libro, dado que se trata de un lugar donde se almacena información con carácter permanente. Tal como no se pueden alterar las palabras de una página impresa, del mismo modo no es posible modificar o suprimir los datos de la ROM

La memoria a corto plazo
La RAM (Random Access Memory: memoria de acceso directo) se parece más a un sistema de archivo que a un libro, puesto que la información se puede modificar y los datos no son permanentes; cuando se apaga el ordenador, la RAM se borra por completo

otros datos. En líneas generales, la memoria RAM es más compleja interiormente que la ROM, porque cada uno de los bits de cada byte de la RAM debe ser capaz de modificarse cada vez que se escribe algo en ellos. Un chip de RAM bastante típico es el Intel 2114. Cada chip de RAM 2114 retiene 4 096 bits de memoria y éstos están organizados en 1 024 *nibbles* (medios bytes) de cuatro bits. Esto significa que cada dirección de localización producirá una salida de 4 bits de datos. Por lo tanto, se necesitarán dos de estos chips para producir un Kbyte completo de información. Cada chip 2114 posee sólo 18 patillas, dos de las cuales se emplean para la toma de tierra y la fuente de alimentación eléctrica. Cuatro se utilizan para las líneas de datos de entrada/salida. Una se usa para la señal de selección del chip (la señal que le indica al chip cuándo se lo requiere o cuándo ha sido "seleccionado") y otra se utiliza para indicarle al chip, una vez que ya ha sido seleccionado, si se está escribiendo en él o si se lo está leyendo. Las diez patillas restantes se emplean para el bus de direcciones. Diez líneas de dirección pueden identificar 2^{10} localizaciones exclusivas, o sea 1 024. Si un ordenador dispusiera de 64 Kbytes de RAM, y si se utilizaran chips Intel 2114, se requeriría un total de 128 chips de RAM, puesto que se necesitarían dos chips para cada Kbyte completo. En la actualidad es más común usar chips de RAM de mayor densidad, que en el mismo espacio alojan más memoria. Utilizando chips de RAM más modernos, como el 4164, se pueden obtener 64 Kbytes de RAM con sólo ocho chips.

Año tras año los chips RAM y ROM van siendo más baratos y más compactos, y en la actualidad es posible que un solo chip contenga 128 Kbits. No obstante, el progreso en cuanto al almacenamiento de densidades aún mayores en un solo chip ya no se produce tan rápidamente como antes. El sistema de circuitos contenido en el diminuto trozo de silicio está llegando a ser tan minúsculo, que las técnicas ópticas que se emplean para "grabar" los circuitos apenas lleguen a ser suficientes para ese trabajo. Los chips de memoria de "gran densidad" del futuro probablemente se fabricarán mediante procedimientos de grabación por haces de electrones o por rayos X.

En líneas generales, existen dos tipos de memoria RAM en uso, conocidos como RAM estática y RAM dinámica. Ambos tipos poseen sus ventajas y sus inconvenientes, pero actualmente se utiliza más la RAM dinámica que la estática. Ambas pierden el contenido de la memoria apenas se desconecta la alimentación eléctrica, pero la memoria dinámica requiere que se le "refresquen" los contenidos cada algunos milisegundos. Cada bit de la memoria se ha de refrescar o reescribir sin entorpecer la capacidad de la CPU para acceder a la información que contengan. Esto significa que se debe diseñar un sistema de circuitos de sincronización específico y muy crítico, lo que dificulta la labor del diseñador de circuitos.

La memoria dinámica presenta dos claras ventajas sobre la memoria estática. La dinámica sólo requiere un transistor por bit, frente a los tres transistores que normalmente necesita la memoria estática para cada bit. Esto permite almacenar más memoria en chips más pequeños. La mayoría de los chips de RAM dinámica poseen sólo 16 patillas. La otra ventaja de las unidades RAM dinámicas es que consumen menos energía que las estáticas. Por lo tanto, generan menos calor y requieren fuentes de alimentación eléctrica más pequeñas y más baratas.

La ventaja de la RAM estática radica en la simplicidad del diseño de su circuito. Una vez escritos los contenidos de la memoria, permanecen en ella sin que sea necesario refrescarlos. Cada celda de memoria de un bit requiere tres transistores, de modo que resulta difícil conseguir las elevadas densidades que permite la RAM dinámica. Asimismo, la RAM estática consume más energía, y el calor extra generado dificulta la labor del sistema de refrigeración del ordenador, lo que puede hacer necesaria la utilización de un ventilador, con el consiguiente encarecimiento del diseño.

Navidad en Basic

Introducimos nuevas órdenes con las cuales se puede manipular la información y escribimos un programa para calcular cuántos días faltan para la Navidad

En este programa pasamos revista a todos los temas que hemos incluido hasta ahora en nuestro curso de programación; asimismo, introducimos algunas sentencias de BASIC nuevas y eficaces. La finalidad del programa es calcular el número de días que faltan para que llegue la Navidad.

Si observa el listado del programa, verá que empieza con una lista de las variables utilizadas. Ciertamente, esta práctica no es esencial, pero sí es aconsejable para que sus programas resulten más fáciles de comprender cuando se encuentre con ellos algún tiempo después. Algunas versiones de BASIC permiten dar a las variables nombres largos, por ejemplo DAY (día), en lugar de una única letra, como hemos venido utilizando hasta ahora. Si usted es afortunado y posee un BASIC que admite variables de nombre largo, escoja para ellas palabras que posean algún significado. DAY (día), MONTH (mes) o DAYNUM (día número) son mucho mejor que A, X o D. Si no tiene alternativa posible porque su BASIC no admite variables de nombre largo, conseguirá casi la misma facilidad de lectura listando las variables al inicio del programa.

Al ejecutar el programa, lo primero que aparecerá en la pantalla serán las sentencias PRINT que comienzan a partir de la línea 230. Éstas describen brevemente lo que hará el programa y a continuación preparan al usuario para que digite la fecha en la forma indicada, usando comas para separar día, mes y año.

La primera sentencia desconocida aparecerá en la línea 300. Es una sentencia DIMension (dimensión). Se utiliza para determinar el número de ítems o de elementos de que consta la matriz denominada X. Una matriz, también llamada variable subíndice, es como una variable común; la diferencia estriba en que su caja contiene diversos compartimientos. En la línea 300 estamos creando una variable denominada X que dentro de su caja posee 12 compartimientos. En un próximo apartado retomaremos el tema de las matrices y la sentencia DIM con más detalle.

```
310 INPUT D, M$, Y
```

Esta línea es una sentencia INPUT normal, sólo que espera tres entradas. D es una variable numérica y contendrá la fecha de hoy. Y, otra variable numérica, corresponderá al año. M$ es algo diferente. Es una "variable alfanumérica", característica que se indica mediante el signo $ (dólar). Una variable alfanumérica acepta tanto caracteres del teclado como números. Si, por ejemplo, digitamos 23, ENERO, 1984, a la variable D se le asignará el valor 23, a la variable M$ la serie de caracteres ENERO y a la variable Y el valor 1984.

```
330 GOSUB 560 REM RUTINA 'N.º DEL MES'
```

Esta sentencia indica que el programa debe bifurcarse hasta la subrutina que comienza en la línea 560. Observe también que en la misma línea se ha incluido una REMark (observación). Si en la línea hay espacio, no siempre es necesario colocar las REM en una nueva línea. Esta subrutina en particular la utiliza el programa una sola vez y, en rigor, podría igualmente haberse incorporado en el programa principal. Al convertirla en subrutina no hemos hecho más que separar esta parte del resto del programa.

Originalmente, cuando este programa se escribió por primera vez, para el mes se utilizó un número y esta parte del programa no era necesaria. Luego se decidió permitir que se diera entrada al mes digitándolo como una palabra completa. Para convertir el mes escrito en palabras a su equivalente en número, se escribió por separado el programa extra que ahora constituye esta subrutina. El único cambio que hubo que introducir en el programa principal (original) fue la adición de una sentencia GOSUB. Esta subrutina ilustra con qué sencillez se pueden estructurar los programas en bloques y enlazar entre sí utilizando las sentencias GOSUB y RETURN.

La subrutina en sí misma es muy sencilla, pero ilustra lo inteligente que es el BASIC para manipular series de caracteres. Supongamos que hemos dado entrada a ENERO como la parte mes de la sentencia INPUT. A la variable M$ se le asignaría, entonces, la serie de caracteres ENERO. La primera línea de la subrutina es:

```
560 IF M$ = "ENERO" THEN LET M = 1
```

Esta sentencia compara el contenido de M$ con los caracteres comprendidos entre las comillas dobles. Si son los mismos (como en este caso), la línea continúa para establecer el valor de la variable numérica M en 1. No confunda la variable M con la variable M$. Son distintas. Sólo una de ellas puede contener una variable alfanumérica: ¡la que posee el signo $! Después de verificar si M$ es la misma que ENERO, el programa se dirige a la línea siguiente y verifica si el contenido de M$ es el mismo que FEBRERO. Si no lo es, M no se establece en 2. El valor de la variable M sólo se establece cuando la correspondencia es correcta y ese valor es el mismo que el número del mes: 1 para enero, 3 para marzo y así sucesivamente.

Al llegar a la línea 680 el BASIC retorna (RETURNs) al programa principal, a la línea posterior de la sentencia GOSUB. Se trata de la línea 340. Contiene una REM pero ninguna observación. Se incluye simplemente para espaciar más el programa y facilitar su lectura.

Entre las líneas 350 y 370 hay un bucle FOR-NEXT. Éste incrementa el valor de I, comenzando por 1 y siguiendo hasta 12. La variable I se utiliza como el subíndice de la matriz X de la línea 360. Se la debe estudiar con suma atención.

```
360 READ X(I)
```

READ es una nueva sentencia que hasta ahora no habíamos visto nunca. READ se utiliza siempre con su correspondiente sentencia DATA. La sentencia DATA para esta línea está en la línea 510:

```
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30,
     31, 30, 25
```

Estos números, a excepción del último, son la cantidad de días que posee cada mes del año. Las dos líneas equivalen a 12 sentencias LET separadas.

```
LET X(1) = 31
LET X(2) = 28
LET X(3) = 31
LET X(4) = 30
LET X(5) = 31
LET X(6) = 30
LET X(7) = 31
LET X(8) = 31
LET X(9) = 30
LET X(10) = 31
LET X(11) = 30
LET X(12) = 25
```

El bucle establecido en la línea 350 hace que I cuente de 1 a 12 para que podamos sustituir X(I), X(1), X(2), X(3), etcétera.

Antes de retomar este programa, consideremos un programa más pequeño y mucho más simple:

```
10 READ A, B, C
20 LET D = A + B + C
30 PRINT D
40 DATA 5, 10, 20
```

Aquí, la sentencia READ de la línea 10 lee el primer ítem de datos (DATA) de la línea 40 y "escribe" su valor en la primera variable. En otras palabras, asigna valor 5 a la variable A. Luego READ lee el siguiente ítem de datos y lo coloca en la variable siguiente. Este programa hace que A = 5, B = 10 y C = 20. Luego suma estos valores y le asigna ese resultado a la variable D. Después este resultado, 35, se imprime (PRINT) en la línea 30.

Volvamos al programa "navideño". La primera vez que se realiza el bucle que comienza en la línea 350, el valor de I se establece en 1. Por lo tanto, la línea 360 equivale a READ X(1). El correspondiente ítem de datos de la línea 510 es 31 (el primer ítem). En consecuencia, X(1) se establece en 31.

La segunda vez que se realiza el bucle, I se convierte en 2, de manera que la línea 360 equivale a READ X(2). El siguiente dato de la línea DATA es 28. Esto significa que X(2) se establece en 28. De este modo, los 12 "compartimientos" de la variable subíndice X se llenan con el número de días de cada mes, excepto el duodécimo, que sólo tiene 25 días. (¿Se ha dado cuenta por qué?)

```
390 GOSUB 750 REM RUTINA 'AÑO BISIESTO'
```

```
100 REM  LISTA DE VARIABLES
110 REM
120 REM  D = FECHA DE HOY
130 REM  M$ = NOMBRE DEL MES
140 REM  Y = AÑO
150 REM  I = INDICE 1
160 REM  X = MATRIZ DE LOS DIAS DE CADA MES
170 REM  R = DIAS QUE FALTAN
180 REM  M = NUMERO DEL MES
190 REM  L = INDICE 2
200 REM  Z = VALOR ENTERO DE Y/4
210 REM
220 REM
230 PRINT "ESTE PROGRAMA CALCULA"
240 PRINT "CUANTOS DIAS FALTAN"
250 PRINT "PARA NAVIDAD"
260 PRINT
270 PRINT "ENTRE DIAS, MES, AÑO DE HOY"
280 PRINT "P. EJ. 12, JULIO, 1984"
290 PRINT
300 DIM X(12)
310 INPUT D, M$, Y
320 REM
330 GOSUB 560 REM RUTINA 'N. DEL MES'
340 REM
350 FOR I = 1 TO 12
360 READ X(I)
370 NEXT I
380 REM
390 GOSUB 750 REM RUTINA 'AÑO BISIESTO'
400 REM
410 LET R = X(M) - D
420 FOR L = M TO 11
430 LET M = M + 1
440 LET R = R + X(M)
450 NEXT L
460 REM
470 IF R = 1 THEN GOTO 500
480 PRINT "FALTAN";R;"DIAS HASTA NAVIDAD"
490 GOTO 520
500 PRINT "FALTA 1 DIA PARA LA NAVIDAD"
510 DATA 31,28,31,30,31,30,31,31,30,31,30,25
520 END
530 REM
540 REM
550 REM
560 IF M$ = "ENERO" THEN LET M = 1
570 IF M$ = "FEBRERO" THEN LET M = 2
580 IF M$ = "MARZO" THEN LET M = 3
590 IF M$ = "ABRIL" THEN LET M = 4
600 IF M$ = "MAYO " THEN LET M = 5
610 IF M$ = "JUNIO" THEN LET M = 6
620 IF M$ = "JULIO" THEN LET M = 7
630 IF M$ = "AGOSTO" THEN LET M = 8
640 IF M$ = "SEPTIEMBRE" THEN LET M = 9
650 IF M$ = "OCTUBRE" THEN LET M = 10
660 IF M$ = "NOVIEMBRE" THEN LET M = 11
670 IF M$ = "DICIEMBRE" THEN LET M = 12
680 RETURN
690 REM
700 REM
710 REM
720 REM NOTA: ESTA RUTINA NO VERIFICA
730 REM       LOS AÑOS BISIESTOS AL FINAL
740 REM       DE CADA SIGLO
750 LET Y = Y / 4
760 LET Z = INT (Y)
770 IF Y - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

que verifica si el año al que se ha dado entrada es bisiesto o no. Veamos cómo funciona.

```
750 LET Y = Y/4
760 LET Z = INT(Y)
770 IF Y−Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

Un año bisiesto es aquel divisible exactamente por 4. Si se trata de un siglo, también debe ser divisible por 400 para que se considere como un año bisiesto. En aras de la simplicidad, no hemos intentado verificar el siglo, sino sólo la divisibilidad por 4.

La línea 750 establece el valor de Y en el antiguo valor de Y (del año) dividido por 4. La nueva Y será un número entero si el año es exactamente divisible por 4. De lo contrario, tendrá una fracción decimal.

La línea 760 utiliza la función INT para hallar el valor "entero" de Y. Si el número al que se ha dado entrada para el año era 1985, el nuevo valor de la función INT se redondeará en los números más aproximados al número entero. El número a redondear se coloca entre paréntesis después de INT. Alternativamente, se puede colocar entre los paréntesis el nombre de una variable. De modo que LET Z = INT (496,25) establecería el valor de Z en 496.

En la línea 770 se resta el valor de Z al valor de Y y se verifica si el resultado es 0. De ser así, significa que el año es un año bisiesto (puesto que la nueva Y no posee fracción decimal). En este caso, el programa se bifurca hacia la línea 790 mediante GOTO. La línea 790 le suma 1 al segundo ítem de la matriz (el segundo ítem era 28, el número de días de un mes de febrero normal).

Si el resultado de la resta de la línea 770 no era 0, X(2) se deja tal como está y la subrutina retorna (RETURN) al programa principal, a la línea 400.

La línea 400 es otra REM utilizada para espaciar el programa y facilitar su lectura. La siguiente línea que verdaderamente hace algo es la 410, donde R es la variable que alberga el número de días que faltan. Aquí se establece, en el número de días del mes al que se ha dado entrada, menos el día al que se ha dado entrada. Si, por ejemplo, hemos dado entrada a 12, FEBRERO, 1984, D sería igual a 12 y M sería 2. Por lo tanto, X(M) sería igual a X(2) y el segundo ítem de la matriz X es 29 (se le ha sumado 1 porque 1984 es un año bisiesto). En consecuencia, R se establecerá en 29 − 12, es decir, 17, el número de días que faltan del mes en curso, febrero.

En la línea 420 empieza otro bucle. Éste está diseñado para incrementar el valor de M. ¿Comprende por qué decimos FOR L = 1 TO 11 en lugar de FOR L = 1 TO 12? Si M fuera 2, porque hubiésemos dado entrada al mes de FEBRERO, la línea 430 la incrementará a 3. Luego la línea 440 establece a R, el número de días que faltan, en la antigua R más X(M). Ésta ahora equivale a X(3), ya que M se ha incrementado en 1. El valor de X(3) es 31, el número de días del mes de marzo. La línea 440, por lo tanto, establece el nuevo valor de R en 17 + 31 (17 era el resultado de haber restado 12 a 29). La próxima vez que se efectúe el bucle, M se incrementará a 4 y se le sumará al antiguo valor de R el número de días de abril, X(4). De manera que la variable R se convertirá en 17 + 31 + 30.

El último circuito a través del bucle se produce después de haberse alcanzado el límite máximo de 11 (en la línea 420). La línea 430 le suma 1 al valor de M por última vez, estableciéndola en 12. El valor de X(12) es el último ítem de la sentencia DATA, 25. Obviamente diciembre no tiene 25 días, pero dado que sólo nos interesa el número de días que faltan para la Navidad, 25 es el número tope del mes.

```
470 IF R = 1 THEN GOTO 500
```

Esta línea simplemente verifica si no falta nada más que un día para Navidad, de modo que obtengamos en pantalla una oración correcta desde el punto de vista gramatical. Si R no es 1 es porque falta más de un día, con lo cual la sentencia PRINT de la línea 480 será gramaticalmente correcta.

Y esto es todo. La versión de BASIC que hemos empleado debería funcionar en la mayoría de los ordenadores (ver el recuadro "Complementos al BASIC"), excepto, quizá, para la subrutina "año bisiesto". El BASIC es muy incoherente en cuanto a la forma de utilizar LET. Si líneas como IF M$ = "SEPTIEMBRE" THEN LET M = 9 no funcionan en su ordenador, puede reescribir la subrutina así:

```
560 IF M$ = "ENERO" THEN GOTO 900
570 IF M$ = "FEBRERO" THEN GOTO 910
580 IF M$ = "MARZO" THEN GOTO 920
.
.
900 LET M = 1
905 RETURN
910 LET M = 2
915 RETURN
920 LET M = 3
925 RETURN
(... y así sucesivamente)
```

Esta solución ocupa más espacio y, con tantos GOTO y RETURN, es más difícil de seguir. Sin embargo, demuestra que por lo general existen diversas maneras de solucionar cualquier problema.

Complementos al BASIC



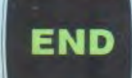
Si el programa se ha de ejecutar en un ordenador BBC, Dragon u Oric, se debe modificar la línea 300 para que diga: 300 DIM X(13)



En el Spectrum no se puede dar entrada a una serie de valores separados mediante comas. Por tanto, se debe modificar la línea 310 de la siguiente manera: 310 INPUT D, y agregar las líneas 312 INPUT M$ y 314 INPUT Y. Como a la fecha no se le da entrada en tres etapas, se deben modificar las sentencias PRINT para que digan: 280 PRINT "¿DIA?", agregando 311 PRINT "¿MES?" y 313 PRINT "¿AÑO?"



Cuando las sentencias REM se colocan al final de otra sentencia, se deben separar mediante dos puntos (:) si se utiliza un BBC, un TI 99/4A o un Spectrum. Por ejemplo, la línea 330 dirá: 330 GOSUB 560: REM RUTINA 'NUMERO DEL MES'



El Spectrum no dispone de sentencia END, de modo que se ha de modificar la línea 520 para que se lea: 520 GOTO 1000, agregando la línea 1000 REM FIN DEL PROGRAMA

Carreras de informática

El profesional de la informática adquiere su experiencia trabajando primero como técnico y ascendiendo poco a poco de categoría



Cortesía de ICL

Gigantes dóciles
Los grandes ordenadores comerciales como el que muestra la fotografía (conocidos como "ordenadores de unidad principal", para diferenciarlos de los mini y microordenadores) requieren de un equipo de operadores altamente experimentados para mantenerse en un nivel de funcionamiento al máximo de sus posibilidades. Las máquinas de estas dimensiones pueden ejecutar cientos de programas simultáneamente y servir a miles de usuarios en cualquier lugar del mundo mediante las líneas telefónicas, los enlaces por microondas y los satélites de comunicaciones

La creciente utilización de los ordenadores en el hogar y en las escuelas está dando lugar a la aparición de muchos programadores geniales, personas que, en otras circunstancias, jamás habrían pensado en la posibilidad de seguir una carrera en el campo de la informática. Pero la cruda realidad es que, como siempre, un aprendizaje superficial es algo peligroso, especialmente si ese conocimiento superficial se limita al lenguaje BASIC. Es importante llegar a comprender que las exigencias de un programador profesional son fundamentalmente diferentes de las de un usuario que programe su ordenador personal, y que muchas de las cualidades no son transferibles.

La elección lógica para una persona que haya completado sus estudios secundarios y tenga un gran interés por los ordenadores sería seguir un curso de informática a nivel universitario o de educación superior, o bien matricularse en una carrera universitaria relacionada con esta nueva disciplina. Muchas universidades y facultades autónomas ofrecen cursos de informática con rango académico, y los estudiantes que hayan obtenido un buen rendimiento al acabar los mismos tienen posibilidades de optar entre diversas ofertas de trabajo. El nivel de desempleo de la industria de la informática se ha limitado al personal informático de nivel inferior, básicamente programadores y operadores, mientras que la demanda de ingenieros, analistas de sistemas y diseñadores no ha disminuido.

Una de las opciones que tiene un campo laboral cada vez más amplio es el de la enseñanza de informática en las escuelas. Hasta ahora, la informática como materia de estudio sólo se había impartido en las universidades y escuelas superiores. En el campo de la educación, el personal capacitado escasea, y una carrera así será muy bien remunerada en el futuro.

En la industria de la informática pueden delimitarse seis niveles principales. El inferior puede describirse como el nivel del "usuario experimentado". Esta categoría incluye a los trabajadores que han aprendido a manejar ordenadores para realizar tareas determinadas, como tratamiento de textos o teneduría de libros. A menudo esta capacitación se interpreta como una experiencia complementaria dentro de otras ocupaciones (por ejemplo, de una secretaria o de un administrativo), pero también comprende funciones de la industria de los ordenadores como son operador de terminales, operador de perforadora de fichas, etc. Para estos empleos se requiere una serie de cualificaciones básicas, bachillerato o escuela superior, y la capacidad de pensar lúcidamente. Experiencias como el dominio del funcionamiento del teclado y otras similares, por lo general se adquieren en el propio trabajo.

El siguiente nivel es el de operador de ordenadores. Aunque los ordenadores que se utilizan en la industria difieren bastante de los ordenadores personales tanto en su aspecto como en su uso, se basan en los mismos principios; por lo tanto, siempre es útil estar ya familiarizados con ellos. Los operadores comprenden muy pronto cómo trabajan los ordenadores y, por ello, ser operador es un buen trampolín para llegar a programador. Considere, no obstante, que esta actividad puede exigirle un intenso esfuerzo físico. Por ejemplo, la mayoría de las grandes instalaciones funcionan 168 horas por semana y durante todo ese tiempo han de contar con un personal que las atienda.

Las principales aptitudes que se requieren para convertirse en programador son poseer una mente lúcida y metódica y capacidad de concentración. Para llegar a ser un programador cualificado se requiere un tipo de aptitud muy especial y, si bien los requisitos normales para la admisión son un título académico o el bachillerato, por lo general es más importante la capacidad para trabajar lógicamente. Los programadores ingresan en la industria sin cualificaciones formales y esta amplitud de miras atrae a muchos padres, esperanzados en las aptitudes de sus hijos para la programación.

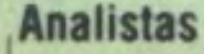
Una elección para toda la vida

Como cualquier otra dependencia de una empresa moderna, el departamento de informática se organiza según unas líneas jerárquicas. A la cabeza del departamento está el **director de proceso de datos,** responsable de las muchas y diversas tareas que han de desarrollarse bajo la denominación general de procesamiento de la información.

Todos los profesionales de la informática se emplean primero como técnicos y obtienen su capacitación a medida que van ascendiendo de categoría. Las tres áreas principales de especialización son: funcionamiento del ordenador, programación y análisis de sistemas; a medida que se avanza en el camino de la promoción, se va adquiriendo un conocimiento básico de las tres áreas de especialización.

Tal como ocurre con otras profesiones, conviene entrar en el campo de la informática con la mayor cualificación posible. Si bien al comienzo parecerá no haber mucha diferencia entre los diferentes grados de cualificación, la persona menos preparada se encontrará pronto con escollos insalvables. ¡Es mucho más difícil concluir una carrera universitaria si uno trabaja durante toda la jornada!

Analistas

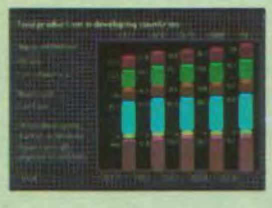




Antes de comenzar cualquier trabajo, se han de estudiar concienzudamente los objetivos y los recursos disponibles. El **analista de sistemas** tiene la función de entrevistar a los usuarios para determinar sus necesidades, conjugar estas necesidades con los recursos disponibles y sugerir un procedimiento para resolver el problema. Para que el analista sea capaz de desarrollar un sistema que funcione para otras personas, debe ser un pensador lógico, con buenas aptitudes para la comunicación y una chispa de creatividad. Con frecuencia es el agente de ventas del departamento de proceso de datos, por lo que debe causar siempre una impresión favorable entre sus "clientes": los usuarios de ordenadores de la empresa.

Programadores

El **programador** toma la amplia estrategia elaborada por el analista y la transforma, primero en un plan táctico, dividiendo el trabajo en segmentos más manejables, y luego en un código que el ordenador pueda reconocer e interpretar. Los **programadores de aplicaciones** se encargan de escribir programas para realizar tareas específicas, mientras que los **programadores de sistemas** están más relacionados con el rendimiento global del sistema de proceso de datos. Los programadores de aplicaciones tienden a trabajar de forma individual, incluso aunque estén integrados dentro de un equipo de proyectos. Para ellos es realmente importante su capacidad para concentrar la atención en la labor que se esté llevando a cabo. Los programadores de sistemas también necesitan de esta capacidad de concentración, pero, además, una actitud reposada. "Si usted puede mantener la cabeza fría cuando a su alrededor todos los demás la están perdiendo...", quizá entonces posea las condiciones requeridas para ser un programador de sistemas.



Ingenieros de desarrollo de sistemas

Aunque tal vez en el futuro sean los propios ordenadores los que desarrollen la nueva generación de máquinas, en la actualidad este proceso de innovación se produce en el cerebro del **ingeniero de desarrollo de sistemas.** Éste desarrolla una labor en parte científica y en parte técnica: consiste en aprovechar los nuevos descubrimientos y desarrollos teóricos para mejorar y ampliar el rendimiento de un componente determinado del equipo. En este campo hasta el profesional menos cualificado ha estudiado cinco años o más en la universidad.



Tony Sleep

Especialistas técnicos

A menudo la única oportunidad de descansar que se le presenta a un operador ocurre cuando el ordenador sufre una avería: entonces se llama a un **especialista técnico** para repararlo. Gracias a la capacidad de los ordenadores modernos para detectar sus propias averías, y a la adopción casi universal de la construcción modular, el trabajo del técnico se ha simplificado considerablemente, si bien éste aún debe ser competente en cuanto a electrónica digital. También ha de ser un mecánico experimentado, capaz de trabajar con tolerancias aún más reducidas que las de un relojero medio. Para acceder a este campo se suele exigir una cualificación de nivel académico.



Operadores

Una de las especializaciones de esta industria que más esfuerzo físico exigen es la de operador de un gran ordenador de multiprogramación-multiusuario. Además de tener que caminar kilómetros en cada turno transportando discos, cintas o cajas de papel, el operador debe estar interiorizado con el sistema operativo del ordenador y con la importancia inherente a cada uno de los trabajos que se ejecuten en la máquina en cada momento.

A un **operador senior** se le puede solicitar que tome decisiones que afecten el trabajo de muchas otras partes de la gestión de la empresa, permitiendo o negando el acceso al sistema de ordenador.

A los **operadores** de instalaciones más reducidas a veces se les solicita que ayuden a los programadores y a los ingenieros para diagnosticar fallos, además de ejecutar uno a uno los programas. Sin embargo, lo más importante es poseer un conocimiento profundo del método operativo del programa. Los programas de "amabilidad hacia el usuario" hacen que el trabajo del operador resulte más sencillo, y un programa bien "preparado" puede ser ejecutado por operadores con relativamente poca experiencia sin pérdida de eficacia.

Kai Choi

El trabajo que requiere menos esfuerzo intelectual es el de **operador de entrada de datos** (perforista-grabador). En este caso, las aptitudes que se requieren son básicamente las mismas que para un mecanógrafo: velocidad y precisión. Lo negativo de este trabajo es que puede llegar a ser repetitivo y tedioso, pero en muchas instalaciones pequeñas esta desventaja se compensa con la oportunidad que ofrece de llegar a familiarizarse con otros aspectos de la actividad del departamento de informática.



Como hemos mencionado anteriormente, comprender el BASIC no implica necesariamente tener abiertas las puertas de la industria informática. A pesar de tratarse de un lenguaje muy popular, la mayoría de los profesionales lo consideran mal estructurado y piensan que favorece hábitos de programación negativos y una forma de pensar muy desordenada. Este problema existe realmente, puesto que la mayoría de los niños con experiencia en ordenadores personales han aprendido BASIC en lugar de otros lenguajes mejor estructurados, como el LOGO o el COMAL.

Las universidades y escuelas universitarias que imparten cursos de informática manifiestan preferencia por la admisión de alumnos *que no hayan aprendido* BASIC, ya que consideran que este lenguaje favorece la adquisición de hábitos muy difíciles de eliminar.

A pesar de este problema, bastantes jóvenes han hallado la manera de ser retribuidos económicamente por sus aptitudes para la programación en BASIC. Muchos de ellos escriben juegos en este lenguaje que les agradan a los otros jóvenes, y las empresas de software están ansiosas por hacerse con juegos dirigidos de forma tan directa a la mentalidad de los adolescentes. Algunos de los "niños precoces" que suelen aparecer en los periódicos porque, a pesar de su corta edad, ganan enormes sumas de dinero, sólo escriben en BASIC y no tienen un conocimiento profundo de la informática. Otros son auténticos fenómenos que escriben en lenguaje Assembly (el lenguaje de bajo nivel que controla con suma eficacia el código de lenguaje máquina de un microprocesador) y tienen por delante un brillantísimo porvenir. Raramente los periodistas están capacitados para diferenciar estas dos clases de jóvenes, y este tipo de reportajes puede inducir a los padres a creer que su pequeño hijo, obsesionado por el ordenador, está preparado para comenzar a ganar dinero a raudales. Esto, aunque posible, es poco probable.

En el negocio

En la industria informática propiamente dicha, los programadores se dividen en dos grupos: los programadores de aplicaciones y los programadores de sistemas. Los primeros escriben programas para realizar una tarea específica. Los programadores de sistemas son "amas de llaves": escriben programas para mantener a punto el sistema de ordenadores, por ejemplo, para detectar averías. El programador de aplicaciones suele encontrarse con otras personas fuera de la sala de ordenadores (clientes) y por lo general trabaja integrado en un equipo desarrollando programas para una función específica. Los programadores de sistemas están más especializados y suelen trabajar en solitario. Ellos hablan directamente con la "inteligencia" de la máquina.

Pero, en este punto, la industria de la informática traza una línea divisoria imaginaria, a partir de la cual sólo permite el acceso a los programadores más brillantes y a los graduados universitarios más cualificados. Éste es el reino reservado a los analistas de sistemas y a los diseñadores.

Los analistas de sistemas consideran un problema y luego deciden cómo puede un ordenador ayudar a resolverlo. Por ejemplo: una compañía petrolera descubre un nuevo yacimiento debajo del lecho marino. Ellos han medido las dimensiones del yacimiento y han comprobado que la calidad del petróleo difiere ampliamente. La compañía petrolera debe decidir si le conviene o no invertir los miles de millones de dólares necesarios para explotar el yacimiento. Esta decisión se tomará de acuerdo con las perspectivas del mercado internacional del petróleo durante el período de vida del yacimiento (supongamos, 20 años), y la compañía debe determinar qué parte del yacimiento ha de perforar primero. Dado que la inversión es tan grande, la compañía petrolera confía el problema a su personal de informática para que lo analice. El analista considera el problema, consulta con economistas, con expertos en marketing de crudos, con geólogos y otros especialistas, y construye un "modelo" por ordenador del yacimiento petrolífero durante los próximos años.

Entonces los ejecutivos de la compañía petrolera juegan con este modelo a "¿qué sucedería si...?", descubriendo cómo incidirían en el rendimiento global diversas decisiones acerca de precio, técnicas de refino y aproximaciones de mercado, recibiendo toda la información que necesitan para tomar las decisiones finales.

En la industria informática existen, además, diversos roles importantes, si bien pocos de ellos se tienen en tan alta estima como el del analista de sistemas. Una excepción quizá sea la del diseñador de hardware. La demanda de ingenieros electrónicos existe a todos los niveles, desde los centros de reparaciones y mantenimiento hasta los departamentos de investigación; pero las áreas de desarrollo del producto de investigación son coto de quienes poseen las más altas cualificaciones en ingeniería electrónica.

En líneas generales existe una notable escasez de personal cualificado en informática. Pero, al mismo tiempo, las cotas de desempleo son igualmente altas, afectando incluso a graduados universitarios y politécnicos. Este evidente desajuste es fuente de preocupación tanto para los responsables de la preparación de personal como para los industriales, y es de esperar que se adopten serias medidas para modificar la situación, incluyendo programas de reciclaje para personas cualificadas en otros campos, y una variedad de oportunidades mucho más amplia para aprender a nivel primario, secundario y terciario.

Diversos gobiernos, particularmente el de Gran Bretaña, consideran que la microelectrónica puede ser la respuesta para algunos de los problemas que plantea el desempleo a corto plazo. El Youth Training Scheme, que pretende proporcionar un entrenamiento y una experiencia laboral a los jóvenes que, habiendo finalizado sus estudios secundarios, todavía no han encontrado su primer trabajo, ofrece actualmente 4 500 plazas en los Information Technology Centres de Gran Bretaña. En estos centros, los jóvenes aprenden diversos aspectos de la microinformática mientras perciben una subvención de capacitación equivalente al subsidio de desempleo. Otros proyectos incluidos en el mismo plan ofrecen la familiarización con el ordenador a aquellos que no llegaron a conseguirla en la escuela (ya sea porque acabaron sus estudios antes de que se iniciara "la era del ordenador", o bien porque no fueron "seleccionados" para utilizarlo), aumentando también sus posibilidades de encontrar un empleo, porque para quien ha dejado el colegio sin haber conocido un ordenador, o para quien es analfabeto informáticamente hablando, las perspectivas de hallar un puesto de trabajo pueden presentarse dudosas.



David Simmonds
David Simmons, de 17 años, el año pasado ganó 10 000 libras esterlinas durante las vacaciones de verano. Es un fenómeno de la programación y escribe programas para la Commodore (que fabrica los ordenadores PET y Vic). A diferencia de muchos adolescentes, David escribe software "serio" para aplicaciones comerciales y, para cuando acabe sus estudios, espera encontrar un puesto lucrativo dentro de la industria informática. David empezó jugando con el ordenador que su padre trajo a casa desde el trabajo, pero muy pronto abandonó los juegos y comenzó a descubrir cómo programar. Inicialmente David publicó algunos de sus programas en la revista que la Commodore edita para sus usuarios, y poco a poco comenzó a vender copias de sus programas. Finalmente la Commodore se enteró de ello y David logró que le dieran la oportunidad de mostrar lo que era capaz de hacer. El resultado fue su primera realización seria en programación



Eugene Evans
Eugene Evans tiene 17 años y percibe unas 40 000 libras al año. Es uno de los pequeños genios que están empezando a aparecer en el campo de la programación de ordenadores y está ayudando a mantener a quien le da empleo, la Imagine Software de Liverpool, entre los primeros fabricantes de juegos para ordenador de Gran Bretaña. Los elevados ingresos de estos fenómenos de la programación generalmente se traducen en concepto de royalties sobre la venta de los juegos (similares a los que perciben los escritores por sus libros), y los adolescentes son los más indicados para desarrollar juegos que agraden a otros adolescentes, principal mercado de los juegos por ordenador

Diagramando la ruta

La cuidadosa utilización de diagramas de flujo lleva a crear programas eficaces y bien organizados

El flujo de información
El objetivo real de un diagrama de flujo es indicar de manera simple y concisa el flujo de información y el control a través de un programa de ordenador.
Los puntos de "condición" son decisivos; hacen que el control se transfiera a un punto diferente del que sigue en la secuencia. Una representación gráfica sencilla de esta transferencia de control es mucho más fácil de comprender que la misma sentencia escrita: realmente, ¡una imagen puede valer más que mil palabras!
El símbolo "CONDICIÓN" puede representarse ya sea mediante un hexágono aplanado, como en la ilustración, o bien mediante una forma de rombo alargado

TERMINADOR
Este símbolo se utiliza para indicar el principio o el final de un subprograma

INPUT/OUTPUT
Representa cualquier proceso que realice una acción interna del programa

PROCESO
Toda entrada o visualización de información (I/O) se representa mediante el símbolo del paralelogramo

CONDICIÓN
Este símbolo se emplea cuando se debe elegir entre rutas de flujo alternativas: VERDADERO o FALSO

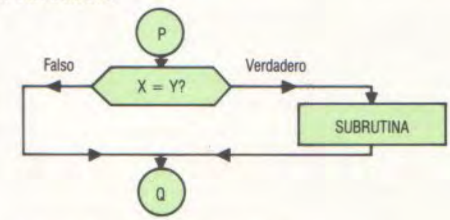
ENLACE
Los círculos indican enlaces entre rutas

Kevin Jones

Un problema puede representarse de forma gráfica sencilla, dibujando diagramas que ilustren los pasos que requiere el procesamiento y los caminos o rutas de flujo que los conectan entre sí. Estos "diagramas de flujo" son útiles como medio para comprender un problema y para trabajar en su solución.

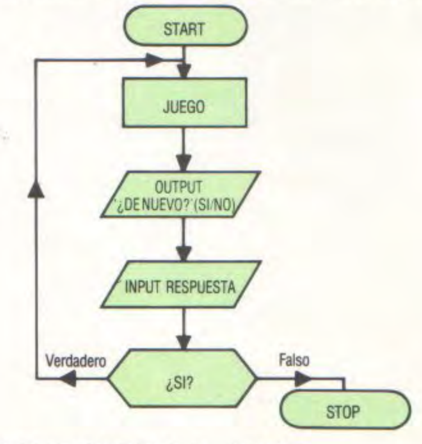
Cada uno de los símbolos de los recuadros en un diagrama de flujo representa un proceso o una acción, y las líneas que enlazan estos recuadros de acción describen los posibles caminos entre ellos. El "flujo de tráfico" es unidireccional, por lo cual se utilizan flechas que indican la dirección, que normalmente es de arriba hacia abajo y de izquierda a derecha del diagrama.

Siempre que haya que elegir una opción, se utiliza un "recuadro de decisión" en forma de rombo o hexágono. Como en el caso anterior, el control fluye por un solo camino, pero puede discurrir en una de dos direcciones, según el resultado de la condición. Si la finalidad de ésta es determinar si un proceso ha de realizarse o no, entonces sólo una de las rutas de salida contiene un "recuadro de proceso". El siguiente ejemplo es una condición para decidir si se deriva o no a una subrutina:




```
120 IF X=Y THEN GOSUB 300
```

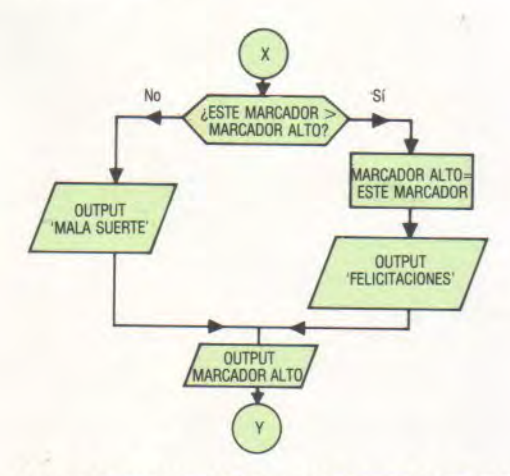
El recuadro de decisión también se usa para indicar la condición que termina un bucle. En el ejemplo que sigue, el control retorna al principio de un programa al dar respuesta positiva a la pregunta '¿DE NUEVO?':




```
90 REM**COMIENZA EL JUEGO**
100
800
```

```
810 PRINT "¿DE NUEVO? (SI/NO)";
820 INPUT R$
830 IF R$ = "Y" THEN GOTO 100
840 END
```

Es posible que deseemos adoptar una decisión que dé por resultado seguir uno de dos cursos de acción claramente diferenciados. En el siguiente ejemplo comparamos el marcador de un jugador con otro marcador anterior más alto:




```
1200 IF ESTE MARCADOR > MARCADOR ALTO THEN
     GOTO 1230
1210 PRINT "MALA SUERTE. DEBE ABANDONAR";
1220 GOTO 1250
1230 LET MARCADOR ALTO = ESTE MARCADOR
1240 PRINT "¡FELICITACIONES! UN NUEVO RECORD";
1250 PRINT MARCADOR ALTO
```

Observe que el valor de MARCADOR ALTO se imprime en ambos casos, y que en el proceso los dos caminos posibles de flujo vuelven a unirse para convertirse en la única entrada para esta operación de salida.

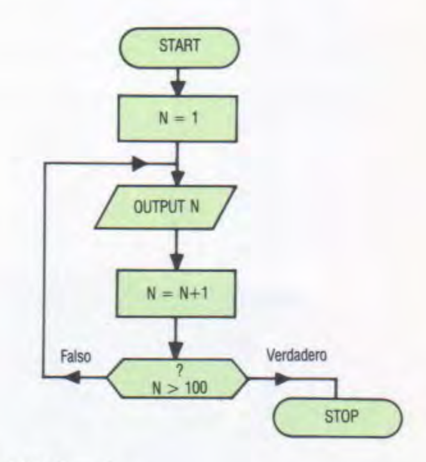
Todas las decisiones se toman como el resultado de condiciones similares a ésta, que dan un resultado positivo o negativo, verdadero o falso. Como puede ver, este proceso de toma de decisiones, puramente binario, niega la posibilidad de una respuesta "quizá". Puede utilizar cualquier término que desee, ¡pero no olvide designar los dos caminos de salida coherentemente!

Todos los lenguajes de programación poseen una sentencia de decisión intrínseca que, si se satisface la condición "verdadero", conduce a una bifurcación condicionada; pero, si el resultado es "falso", deja que el control pase a la sentencia siguiente. En el caso de una versión de BASIC que permita sólo un IF-THEN simple, debemos imitar la bifurcación condicionada mediante una sentencia GOTO, como en la línea 1200 del último ejemplo. La sentencia de la línea 1210 sólo se ejecutará si el resultado de la condición de la línea 1200 es falso.

¿Y qué sucede con la segunda utilización de GOTO en la línea 1220? Como puede observar, el empleo de GOTO al final de la condición, para solucionar el problema del destino de la bifurcación condicionada, nos ha obligado a usar este procedimiento para volver a "unir" los dos caminos de control posibles, en este caso en la línea 1250.

Normalmente la utilización de diagramas de flujo favorece la introducción de sentencias GOTO como recurso para seguir punto por punto la representación gráfica del programa. En general, esta forma de utilizar los saltos no condicionados resulta bastante peligrosa. Si la versión del BASIC que se está empleando obliga a esta solución, el diagrama de flujo constituye un método excelente para que estemos en condiciones de juzgar en qué forma el control sigue la sucesión normal del programa.

Tomemos un último ejemplo para estudiar cómo la utilización de un diagrama de flujo nos permite representar exactamente los pasos necesarios para realizar una tarea sencilla: imprimir todos los números comprendidos entre uno y cien.




```
10 LET N = 1
20 PRINT N
30 LET N = N+1
40 IF N > 100 THEN END
50 GOTO 20
```

Esta forma de utilizar los diagramas de flujo tiende a favorecer un enfoque paso a paso de la escritura del programa, lo cual con frecuencia da como resultado programas poco elegantes, particularmente en proyectos más largos. Para quienes poseen un conocimiento aunque sea superficial del lenguaje BASIC, obviamente es más conveniente emplear un bucle FOR-NEXT. Por ejemplo:

```
10 FOR N = 1 TO 100
20 PRINT N
30 NEXT N
40 END
```

En un diagrama de flujo es imposible representar este BASIC "taquigráfico" y, de seguirlo exactamente, nos llevaría a una forma menos eficaz de resolver el problema. No obstante, nos proporciona cierta información acerca de la estructura del bucle FOR-NEXT y, por lo tanto, será valioso, cuando examinemos ésta y otras funciones BASIC, para determinar cómo están construidas.

Los diagramas de flujo son particularmente útiles durante la etapa de planificación o conceptualización de la programación, especialmente en las partes "difíciles". Los programadores muy experimentados tienden a utilizarlos menos que los principiantes, y a menudo recurren a un diagrama de flujo para ilustrar y documentar un programa escrito sin su ayuda. Sin embargo, tanto si el diagrama de flujo se traza sobre papel como si sólo existe en el subconsciente del programador, la idea de diagramar el flujo de información y de control es esencial para el usuario de un ordenador como herramienta para resolver los problemas.

Ciencia hogareña

Si cree que en su hogar no hay ordenadores, deberá mirar con mayor atención...

¿Cuántos chips de microprocesador hay en su hogar? Por supuesto, puede haber uno en el corazón de su ordenador personal; pero, ¿y la lavadora, el equipo de alta fidelidad o, quizá, el video?

Todo, desde el horno hasta el encendido del coche y su tablero de instrumentos, puede alardear de poseer un chip.

No olvide la calculadora, arrinconada en un cajón del escritorio, o su reloj digital: ¡son los primeros ejemplos de fabricación masiva de microprocesadores!

Juego para los niños
En el hogar, los niños suelen utilizar los ordenadores más que los adultos, familiarizándose con ellos con la misma naturalidad que si se tratara de un televisor. Conociendo la tortuga y el lenguaje LOGO, los niños aprenden a explorar y a satisfacer su curiosidad por sí mismos. Incluso un LOGO sencillo, que sólo posea gráficos tortuga, puede ayudar al niño más pequeño del hogar. También existe software educativo de gran calidad (véase p. 81). Los juegos pueden estimular y educar, pero con frecuencia el interés y el talento de un niño se desarrollan y se refuerzan mucho más enfrentándolo a los problemas que supone programar realmente un ordenador. Actualmente el LOGO destinado a muchos ordenadores personales se está extendiendo y abaratando y ofrece un enorme potencial para favorecer la mejor forma de abordar la solución de un problema en muchos campos, no sólo en la programación de ordenadores

Steve Cross

El chip que lava
Algunas lavadoras utilizan un microprocesador para seleccionar y supervisar los diversos ciclos de lavado y rotación que se requieren para tratar adecuadamente todos los tejidos susceptibles de lavarse a máquina. Se pueden visualizar y seleccionar al tacto las mejores combinaciones de acciones de lavado y temperaturas, niveles de agua, proceso de aclarado y velocidades de centrifugación. Puesto que no existen otros componentes móviles más que el tambor, el período de vida útil de la máquina es muy largo y los costos de mantenimiento se reducen considerablemente

El micro móvil
Los coches están empezando a incorporar ordenadores para mejorar el rendimiento y proporcionar nuevas prestaciones. Éstos pueden calcular el consumo de gasolina, controlar la velocidad, actuar como alarma antirrobo e incluso hablar con el conductor para advertirle que la presión del aceite es muy baja o que se está agotando la batería

El chip de coser
La máquina de coser tradicional puede realizar puntadas seguras y maravillosamente parejas, pero para ello se requiere paciencia y experiencia por parte del usuario. Una máquina de coser controlada por microprocesador puede ayudar a crear con muy poco esfuerzo un complicado patrón de bordado o una puntada delicada. Además de la selección de clases de puntadas preestablecidas, estas máquinas de coser se pueden programar para producir otras costuras, diferentes de las almacenadas en la memoria incorporada, aun cuando estén desconectadas

Su ordenador personal
Es probable que al principio usted haya utilizado su ordenador personal sólo para jugar; pero ¿qué le parecería destinarlo a actividades que supongan un mayor desafío? Si dirige una empresa o si es un padre interesado en la educación de sus hijos, es probable que ya esté utilizando su ordenador personal de muchas maneras: llevando el estado de sus cuentas o aprovechando el valor educativo del ordenador. Sin embargo, existen una serie de aplicaciones que tal vez no se le hayan ocurrido. Le interesará saber, por ejemplo, que el grado de sofisticación que puede introducir en un sistema de seguridad es ilimitado. El ordenador puede supervisar sensores ocultos de diversos tipos, disparar alarmas, e incluso telefonear automáticamente a los servicios de emergencia

Compensación completa
Los microprocesadores pueden proporcionar un control de color muy estable del sistema de circuitos del televisor. Automáticamente compensan una desviación de sintonía, las variaciones de temperatura y el envejecimiento de los componentes. Pronto se emitirán servicios de teletexto. En Gran Bretaña, el servicio Ceefax en los dos canales de la BBC y el Oracle en los canales de la ITV dan acceso a una vastísima gama de informaciones que se hallan almacenadas en una base de datos. Esta información se actualiza constantemente, de modo que en cualquier momento y de forma instantánea se pueden conocer las últimas novedades en cuanto a la actualidad, tiempo, deportes e incluso las cotizaciones de la bolsa

Grabadora de video
Para mantener el color, la definición y la estabilidad de la imagen, una cinta de video ha de grabarse y reproducirse con una sincronización exacta. El control de precisión del mecanismo de cinta en un video es una tarea muy adecuada para el poderoso microchip con el que cuentan algunos modelos. También puede encargarse de grabar programas de televisión cuando usted no esté en casa. Basta con programar el sintonizador automático con la hora y los canales que le interesen

Controlando por reloj
Los sistemas de calefacción central pueden controlarse de forma más cabal y eficaz mediante un microprocesador que a través de los métodos convencionales. El reloj electrónico del chip permite programar convenientemente las diversas necesidades de calefacción durante los días laborables y los fines de semana. Zonas separadas como los dormitorios o el garaje también pueden tener sus propios programas de control de tiempo y temperatura. Un control como éste ahorra energía y reduce las facturas

No más comidas quemadas
Un horno informatizado puede ayudarle a preparar platos perfectamente cocinados mediante el control exacto del tiempo y de la temperatura. Mientras el plato se cocina según la temperatura y el tiempo programados, la siguiente receta aparece en pantalla a través del adaptador para teletexto del ordenador personal

El eslabón perdido

Por medio del modem, la información puede pasar de un ordenador a otro a través de miles de kilómetros



Ian Dobbie

El acoplador acústico
La mayoría de los modems que existen en la actualidad son dispositivos "totalmente electrónicos" que conectan directamente con la línea telefónica y se enchufan en el conector del teléfono. Las compañías telefónicas obligan a respetar unas pautas muy rígidas para los dispositivos del tipo de los modems, y ésa es la razón por la cual tienden a ser tan caros. Una solución más económica, puesto que pasa por alto la reglamentación en este sentido, consiste en utilizar un "acoplador acústico". Se trata de una clase de modem que convierte las señales de audiofrecuencia por ondas sinusoidales en sonidos verdaderos que alimentan un pequeño altavoz

La palabra *modem* es la contracción de "modulador /demodulador". A pesar de que ya hace aproximadamente cinco años que los modems se venden en el mercado, sólo ahora está comenzando a aumentar el número de propietarios de ordenadores personales que los utilizan. Si todos podemos ser propietarios de un ordenador, ¿qué sentido tiene gastar dinero en comprarse un modem para conectarlo con el sistema telefónico?

Con un modem, su ordenador podrá «hablar» con otros ordenadores situados en cualquier lugar del mundo. Para ello lo único que se requiere es que el ordenador que se encuentre en el otro extremo de la línea telefónica también posea su propio modem. Este otro ordenador puede ser tan sólo un corriente micro personal perteneciente a otro entusiasta de la informática, o bien un gran ordenador de unidad principal propiedad de una universidad o de una institución financiera. Conectar su ordenador con un gran ordenador de unidad principal puede permitirle el acceso a grandes bancos de datos, servicios de información e incluso a las últimas cotizaciones de la bolsa. Al conectar su micro al de un amigo podrían ambos intercambiar software o enviarse un "correo electrónico" gratuito, e incluso jugar en dos direcciones.

Los modems funcionan de forma similar a la interface para cassette que acompaña a la mayoría de los ordenadores personales. Tanto las interfaces para cassette como los modems convierten los unos y los ceros del ordenador en frecuencias audibles. En el caso de las interfaces para cassette, estas frecuencias se pueden grabar fácilmente, como si fueran señales de audio en la cinta de cassette. En el caso de los modems, las frecuencias audibles se envían a través de la línea telefónica y son convertidas en números binarios por el modem situado en el otro extremo.

Sin embargo, para grabar en cinta las interfaces para cassette sólo necesitan convertir las señales binarias en señales audibles (este proceso se denomina *modulación*). O bien realizan lo contrario y convierten las



La máquina FAX
Las máquinas FAX (abreviación de máquinas facsímil) se están popularizando rápidamente en las oficinas de Europa y de Estados Unidos. En Japón hasta las empresas más pequeñas las poseen, y también se utilizan en muchos hogares. Las máquinas FAX pueden transmitir grandes documentos, incluyendo dibujos e imágenes, a otras máquinas FAX en cuestión de segundos, valiéndose sólo de un modem incorporado y de un teléfono corriente

señales audibles reproducidas por la cassette en señales binarias *(demodulación)*. Por otra parte, la mayoría de los modems están diseñados para comunicarse bidireccionalmente a través de una única línea telefónica y, por tanto, requieren dos bandas de frecuencias y cuatro frecuencias individuales. Un modelo popular usa una frecuencia de 1 070 Hz para un 0 y de 1 270 Hz para un 1 para transmitir, y 2 025 Hz para un 0 y 2 225 Hz para un 1 para recibir. Observará que las dos frecuencias en cada una de las bandas (de baja frecuencia y de alta frecuencia) están muy próximas. Sólo hay una diferencia de 200 Hz de frecuencia para un 1 y un 0 en ambas bandas. Esto contrasta con las interfaces para cassette, en las cuales la frecuencia para un 1 es normalmente el doble de alta que la frecuencia para un 0. Para decodificar frecuencias tan próximas se requiere un sistema de circuitos electrónicos muy complicado y por ello los modems pueden considerarse un lujo: un modem puede costar tanto como varios micros personales pequeños.

Atari 400 y 800

Los juegos constituyen el punto fuerte de la gama de ordenadores Atari



Chris Stevens

La prosperidad de Atari, que ahora tiene sus oficinas centrales en Sunnyvale, Silicon Valley (California), es consecuencia del fabuloso éxito que ha obtenido con sus juegos recreativos. El primero de éstos fue el "Pong", en blanco y negro, que se jugaba en la pantalla del televisor.

Después de este primer y humilde intento, Atari fue creciendo y finalmente se integró en el gran Warner Communications Group; en la actualidad, seis años después, es uno de los primeros fabricantes de ordenadores personales, además de poseer una parte muy grande del mercado de juegos recreativos.

Con sus excelentes normas de construcción y la seguridad que transmite su sólido aspecto, la gama de ordenadores personales Atari, que consta de los modelos 800 y 400, ha sentado unas pautas que otros fabricantes intentan emular. Los productos Atari gozan de una imagen de gran calidad, y a ello contribuyen sus soberbios gráficos y su variado software.

El modelo 400, cuyo precio ronda las 80 000 pesetas, difiere del modelo más caro (el 800 cuesta alrededor de 166 000 pesetas) en su memoria máxima (de 16 Kbytes el 400, ampliables a 48 Kbytes en el 800), en que sólo tiene una puerta para cartuchos en lugar de dos, y, aunque quizá se trate de una característica menos relevante, en que su uso está limitado a un televisor corriente, cuando el modelo 800 permite también la visualización vía monitor. Sin embargo, la diferencia más evidente y tal vez la más trascendente se encuentra en el teclado.

Para solucionar el problema de los niveles de señal inarmónicos, los ordenadores personales Atari no utilizan grabadoras de cassette corrientes, sino que requieren (en sus dos versiones) una especial de la misma marca, que alcanza una velocidad de transferencia de datos de 600 bits por segundo. La capacidad de almacenamiento es de 100 Kbytes en una cinta de 60 minutos.

Los tecladós Atari
La diferencia más evidente entre estos dos modelos Atari radica en sus teclados. Mientras que el modelo más grande, el 800, está equipado con un teclado completo similar al de una máquina de escribir, el modelo más pequeño posee un teclado de tipo membrana que, si bien es mejor que otros, adolece sin embargo de los defectos de otras unidades semejantes; básicamente, de una falta de "sensación" y, algunas veces, una respuesta impredecible. Pero algunas máquinas caras y funcionales, fabricadas con fines industriales y militares, utilizan este método ¡para evitar la acumulación de polvo y el ocasional derramamiento de una taza de café!





La unidad de disco
Considerado como el periférico más útil, en la actualidad el Atari 810 está empezando a evidenciar su edad. Con sólo 88 Kbytes por disco resulta más bien pequeño y, como se conecta con la máquina a través de una interface en serie, también resulta lento. Sin embargo, tiene un sofisticado sistema operativo con muchas configuraciones derivadas de otros programas

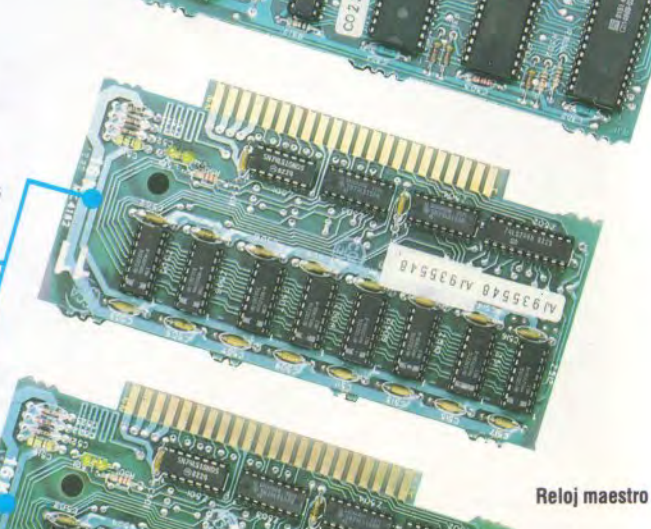


La unidad de cassette
Tratándose de una unidad especial diseñada para funcionar con los ordenadores Atari, la unidad de cassette Atari 410 es más fiable y más sencilla de operar que las habituales unidades disponibles. Por el mismo motivo, no posee altavoz, con lo cual consigue una reducción de peso y de tamaño, así como tampoco puede reemplazarse por una cassette corriente. Su velocidad de transferencia de datos es de 600 bits por segundo



La palanca de mando Atari
La palanca de mando Atari es uno de los accesorios más pobres de la máquina. Es de tipo bastón, por lo cual no ofrece variabilidad. Sólo proporciona un impulso en la dirección requerida y es bastante rígida

PLACA DE LA CPU



ANTIC
El ANTIC, uno de los chips especializados que otorgan al Atari sus impresionantes configuraciones, controla el movimiento en espiral de la pantalla, el lápiz óptico y una de las interrupciones

CTIA o GTIA
Este chip, exclusivo del Atari, manipula el color, algunas I/O heterogéneas y los gráficos de los juegos bélicos

Reloj de color
Mediante este mando se controla el color. Las diversas resoluciones para gráficos se seleccionan haciendo variar la velocidad de este reloj

CPU 6502

Ajuste de color

PLACAS DE LA RAM

El Atari puede disponer de hasta tres placas de circuitos impresos, que permiten ampliar la memoria hasta 48 Kbytes

Reloj maestro

POKEY
Tercero de los chips fabricados a medida, controla el teclado, I/O en serie, los sincronizadores del sistema y también el sonido

Adaptador para interface periférica 6520
Este chip vigila los controladores manuales

Altavoz

PLACA DE PERSONALIDAD

El Atari puede convertirse en una máquina bastante diferente si se reemplazan estas ROM por otras. Por ejemplo, pueden utilizarse lenguajes alternativos





PLACA MADRE

Diodos rectificadores
La fuente de alimentación eléctrica del Atari produce una salida de corriente alterna, y estos componentes la convierten en corriente continua

Conexión periférico

Interruptor de potencia On/Off

Ranuras ampliación 1, 2 y 3
En estas conexiones, cada una de las cuales amplía la máquina en 16 Kbytes, puede enchufarse un cartucho RAM

Selector de canal

Conexión CPU
Dado que el Atari posee la CPU en una ficha separada, esta ranura es para albergarla

Conexión señal de video

Enchufe red

Ranuras para cartuchos
Ambas (izquierda y derecha) son para cartuchos ROM preprogramados con juegos o programas útiles

Ranura para ROM

Interruptor de protección
Desconecta la alimentación eléctrica cada vez que se levanta la tapa, para evitarle riesgos al usuario

Conexión de salida del monitor

Int. de comienzo

Interruptor de selección

Interruptor del sistema

Interruptor facultativo

Conexión para teclado

LED indicadores de potencia

Conexiones para controladores manuales

ATARI 800

DIMENSIONES
405 × 330 × 110 mm

PESO
4 200 g

CPU
6502

VELOCIDAD DEL RELOJ
1,79 MHz

MEMORIA
16 K a 48 Kbytes

VISUALIZACION EN VIDEO
Texto en pantalla: 40 × 24 caracteres; gráficos en pantalla: máximo de 329 × 192 puntos, incluyendo 16 colores y 8 matices

INTERFACES
Conexión televisor, monitor, grabadora de cassette (unidad especial), 4 palancas de mando, conexión serial

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
BASIC A+, PILOT, C

VIENE CON
Fuente de alimentación eléctrica (sin enchufe), manual

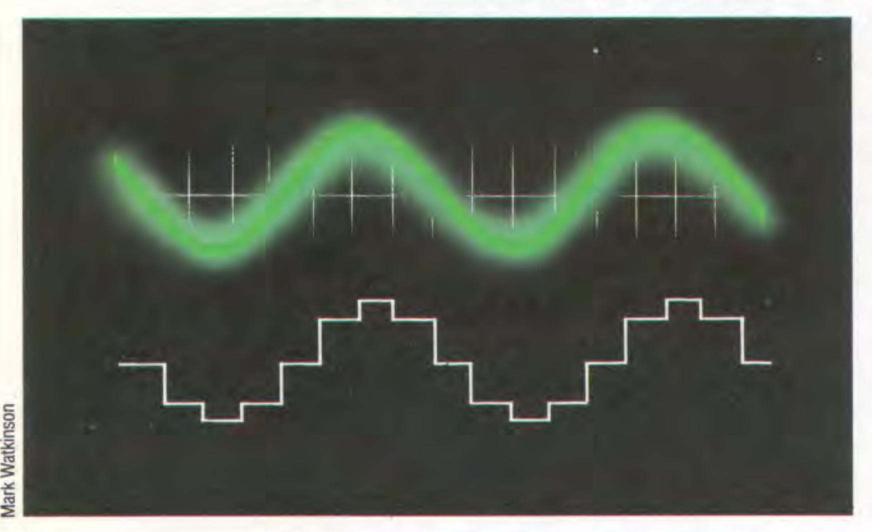
TECLADO
57 teclas móviles individuales, más 3 teclas de función

DOCUMENTACION
Los manuales de iniciación están escritos con gran claridad y su elaboración es correcta y precisa. Atari proporciona notas técnicas más completas para los usuarios más experimentados. Este avanzado manual es el mismo que utilizan los ingenieros de la Atari, y no sólo contiene el diagrama de todo el circuito, sino también los listados de muchos de los programas que controlan el funcionamiento interno del ordenador (el software del sistema). El único defecto del manual es su formato; viene en páginas sueltas para una carpeta de tres anillas que, no obstante, no se proporciona

Diálogo digital

Input/Output, o Entrada/Salida: esenciales para el funcionamiento de cualquier sistema de ordenadores

De analógico a digital
En el mundo real, es muy poca la información que llega con pasos digitales, discretos. Por el contrario, la mayoría de ella es tan variable como los niveles de ruido o las mareas.
Con el fin de que esta información le resulte comprensible al ordenador, la señal ha de ser primeramente digitalizada. El convertidor Analogue-to-Digital, A/D (de analógico a digital), toma muestras de la fuente de señales a una velocidad constante y conocida, tal vez un centenar por segundo. Cada una de estas muestras se almacena en una dirección separada de memoria en forma de valor digital, permitiendo, de este modo, que se realicen los cálculos de variación y se reconozcan las condiciones fuera del límite.
Los convertidores Digital-to-Analogue, D/A (de digital a analógico), realizan una función similar pero a la inversa y para suavizar los picos y obtener una curva regular se utilizan técnicas estadísticas


Mark Watkinson

Input/Output (Entrada/Salida) o, como se abrevia comúnmente, I/O, es el término que se emplea para describir la transferencia de información entre la CPU (Unidad Central de Proceso, que constituye el corazón del ordenador) y el "mundo exterior". "Mundo exterior" equivale, en este contexto, a todos los dispositivos que pueden conectarse al ordenador. No incluye ni la memoria RAM ni la memoria ROM, que se consideran como integradas en el ordenador. La distinción entre lo que sucede en el "interior" del ordenador y lo que ocurre en el "exterior" es un tanto arbitraria. Pero todos los circuitos lógicos (véase p. 92), diseñados para trabajar en íntima unión con la CPU y la memoria principal, se consideran como pertenecientes al "interior" del ordenador.

Los dispositivos externos, que utilizan I/O para comunicarse con el ordenador, incluyen una gran variedad de periféricos, desde el teclado a las unidades de disco flexible, pasando por las palancas de mando, las impresoras y las unidades de visualización de video.

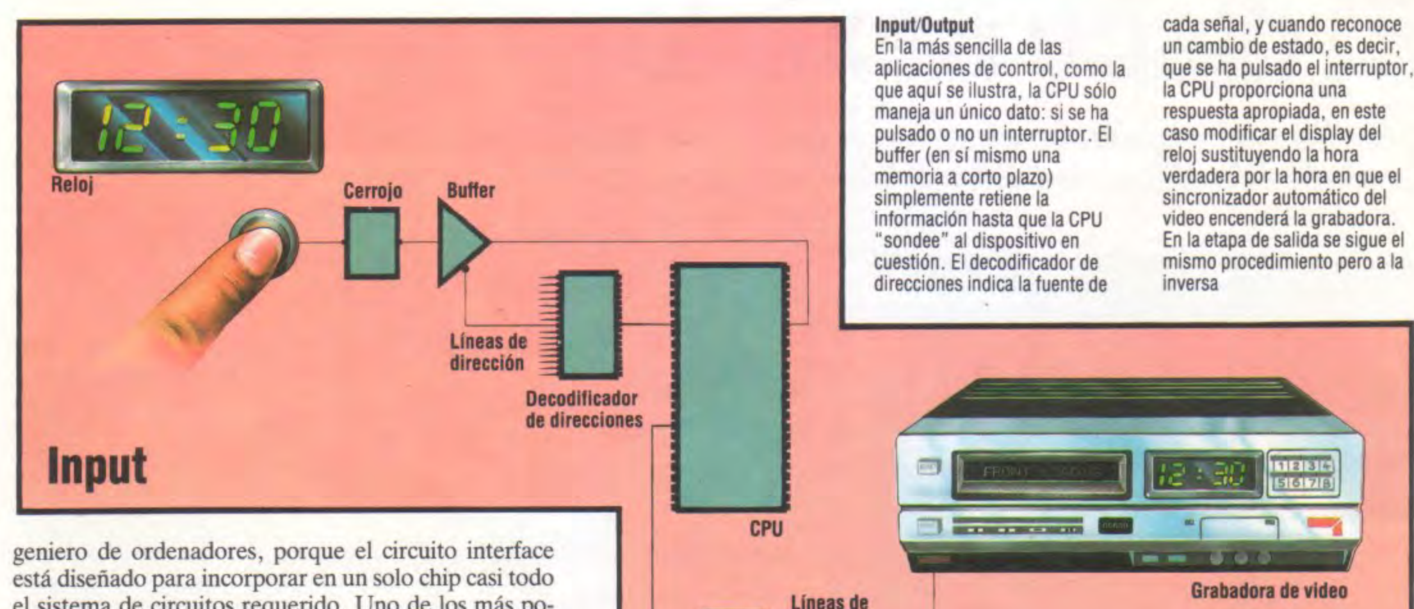
Cuando la CPU desea recuperar información de su memoria, primero debe "localizar" la dirección en donde se halla almacenado el byte de información. Del mismo modo, si la CPU desea almacenar un byte de información para utilizarlo posteriormente, debe primero localizar la dirección en donde se ha de almacenar el byte. Este proceso se denomina *direccionamiento de memoria*. Implica la colocación, por parte de la CPU, de los dígitos binarios correspondientes a la dirección de memoria deseada en una serie de 16 cables conectados a las "patillas de dirección" de la CPU. Estos cables se denominan *bus de direcciones*. Un sistema de circuitos especial en la sección de memoria decodifica estos 16 dígitos binarios para seleccionar la dirección de memoria correcta. (Dieciséis dígitos binarios pueden dar 65 536 combinaciones exclusivas de unos y ceros y, por tanto, localizar la misma cantidad de direcciones de memoria diferentes.)

Si el ordenador desea comunicarse con un dispositivo externo, debe localizar ese dispositivo de modo similar. Sólo dispone de ocho líneas de dirección. Esto limita a 256 el número total de direcciones de I/O separadas que pueden seleccionarse. Es una cantidad pequeña comparada con el potencial de direccionamiento de 16 líneas de dirección, pero en la práctica 256 resulta más que suficiente. Normalmente no se plantea la necesidad de conectar a un ordenador una cantidad mayor de unidades externas.

Seleccionando los dispositivos

Para averiguar cómo actúa realmente el ordenador para seleccionar una unidad externa y enviarle información, consideremos uno de los dispositivos de salida más sencillos: un LED (Light Emitting Diode: diodo emisor de luz) montado en el teclado del ordenador para indicar cuándo se ha pulsado la tecla CAPS LOCK (el microordenador BBC posee una tecla y un LED de este tipo). Para el ordenador, el LED es tan sólo otro dispositivo externo al cual puede enviarle información. En el caso de un único LED, la información será un único 1 (para encenderlo) o bien un único 0 (para apagarlo). Aun tratándose de un LED modesto que requiera una única información, necesita una dirección o localización. La CPU no puede ocupar todo su tiempo direccionando un LED. Necesita poder seleccionarlo una sola vez para decirle cuándo ha de encenderse, y otra vez para comunicarle cuándo ha de apagarse. Supongamos, sólo para seguir con el mismo razonamiento, que la dirección I/O del LED es 32. Para seleccionarla, las líneas de dirección serán, para la CPU, el equivalente binario de 32. Esto es, 00100000 en binario. El LED tendrá un circuito "decodificador" especial que ignorará todas las otras combinaciones de bits en las líneas de dirección. Cuando la línea de dirección sea 00100000, este circuito decodificador la reconocerá y producirá un voltaje alto y, por lo tanto, una salida "verdadera". Lo siguiente que se requiere en el circuito para hacer que el LED se encienda, es un pequeño chip llamado *cerrojo de datos*. Este chip cierra o retiene los datos que se le envían, de modo que el LED permanece encendido o apagado hasta la próxima vez que el chip es direccionado y se le envía una nueva información. Este proceso recibe el nombre de *toggling*.

La mayoría de los dispositivos externos con los que se comunica el ordenador son bastante más complicados que un LED. La impresora es un periférico típico y cada vez que el ordenador se comunica con ella los datos transmitidos representan el código para que se imprima un carácter. Normalmente, cuando han de transferirse grandes cantidades de información, como en el caso de una impresora, se utiliza un chip interface I/O especial. Estos chips simplifican la labor del in-



Input/Output
En la más sencilla de las aplicaciones de control, como la que aquí se ilustra, la CPU sólo maneja un único dato: si se ha pulsado o no un interruptor. El buffer (en sí mismo una memoria a corto plazo) simplemente retiene la información hasta que la CPU "sondee" al dispositivo en cuestión. El decodificador de direcciones indica la fuente de cada señal, y cuando reconoce un cambio de estado, es decir, que se ha pulsado el interruptor, la CPU proporciona una respuesta apropiada, en este caso modificar el display del reloj sustituyendo la hora verdadera por la hora en que el sincronizador automático del video encenderá la grabadora. En la etapa de salida se sigue el mismo procedimiento pero a la inversa

geniero de ordenadores, porque el circuito interface está diseñado para incorporar en un solo chip casi todo el sistema de circuitos requerido. Uno de los más populares es el PPI 8255 (Programmable Peripheral Interface: interface periférica programable). Este chip de 40 patillas contiene tres puertas de I/O de ocho bits. Esto significa que en el chip hay 24 patillas I/O, ocho para cada una de las puertas de I/O, A, B y C. Cada una de estas puertas puede enviar ocho bits (el valor de un byte) de datos a la vez a un dispositivo periférico, como una impresora, o recibir ocho bits de datos a la vez desde un dispositivo de entrada, como, por ejemplo, un teclado.

Para enviar ocho bits de datos a una impresora, la CPU se dirige primeramente al PPI y luego le envía ocho bits de información en el bus de datos. Esta información se almacena provisionalmente en la celda de memoria de un byte situada en el chip, que se conoce como registro. El PPI, entonces, hará que esta información quede disponible en el juego apropiado de patillas I/O. Un principio similar pero que funciona a la inversa permite almacenar los datos provenientes de dispositivos de entrada externos en un registro del chip, y luego colocarlos en el bus de datos cuando la CPU le envía la señal adecuada. Como hemos mencionado previamente, no se puede permitir que los dispositivos externos coloquen su información continuamente en el bus de datos del ordenador; se necesita transferir información desde y hacia la memoria y por otros dispositivos de I/O. El chip de I/O almacena temporalmente la información y sólo la coloca en el bus de datos (para que sea recogida por la CPU) cuando la CPU le indica que lo haga.

¿Cómo sabe la CPU que un dispositivo externo está intentando enviarle información al ordenador? Brevemente, existen dos procedimientos principales. La CPU deja de ejecutar el programa que está llevando periódicamente y echa una rápida "mirada" a todas las puertas de entrada. Si allí descubre que hay una información que está esperando entrada, instruye a la puerta en el sentido de que coloque la información en el bus de datos. El proceso de investigación de los dispositivos de entrada se denomina *sondeo*.

El otro procedimiento se basa en "interrupciones". El dispositivo a la espera de atención envía una señal de interrupción directamente a la CPU, que obliga a detener el programa que se está ejecutando mientras se atiende a la puerta de entrada. Más adelante nos ocuparemos con mayor detalle de las ventajas y los inconvenientes que ambos procedimientos ofrecen al usuario.

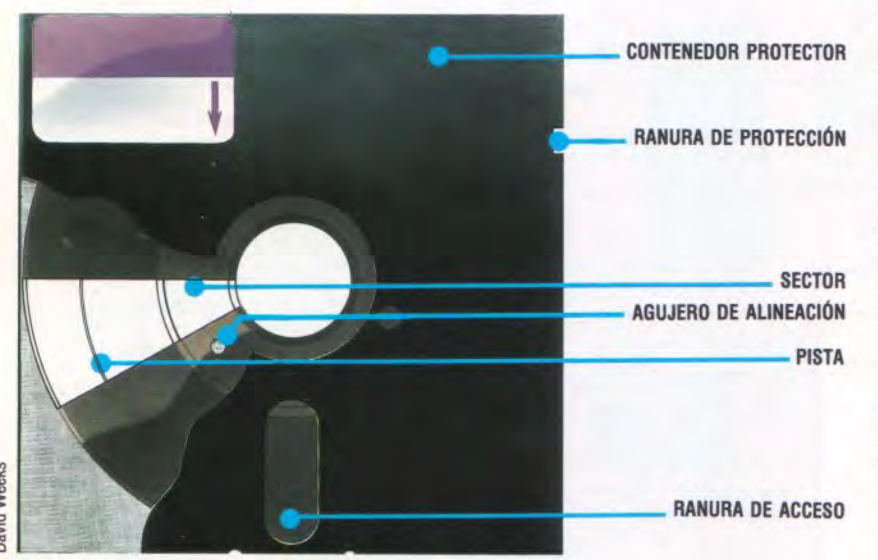
La I/O que hemos descrito hasta ahora se denomina *I/O en paralelo*, porque la información sale o entra a razón de un byte a la vez utilizando ocho cables o líneas de I/O (ocho bits en paralelo). Existe otra técnica que recibe el nombre de *I/O en serie*. En este caso la información de cada byte se alimenta a razón de un bit a la vez, uno detrás de otro. Algunas impresoras emplean interfaces en serie, y la salida de los modems (véase p. 108) también se realiza en serie. La principal ventaja radica en que, básicamente, la comunicación en serie permite usar un solo par de cables en lugar de ocho o más.



Puertas en serie y puertas en paralelo
La mayoría de los microordenadores modernos poseen puertas tanto en serie como en paralelo; las primeras pasan los datos de a un bit a la vez, mientras que las segundas lo hacen en forma de bytes. El tipo de convención en serie más común, conocido como RS232C, puede utilizar una conexión subminiatura de "tipo D", como el ejemplo de 25 patillas que muestra la fotografía (izquierda), o bien, aunque más raramente, un enchufe DIN como el empleado en los sistemas de alta fidelidad.
La puerta en paralelo (derecha) responde a la convención IEEE488, desarrollada por la Hewlett Packard y que el Institute of Electrical and Electronics Engineers de Estados Unidos ha adoptado como estándar para toda la industria

El disco flexible

Los discos magnéticos giran a gran velocidad en el interior de las unidades de disco, llevando información que el ordenador "lee"



David Weeks

El floppy o diskette
La superficie de un disco flexible está dividida en una cantidad de bandas separadas denominadas pistas. Estas pistas, a su vez, están subdivididas en sectores. En el Apple II, por ejemplo, cada pista se divide en 16 sectores. Cada sector posee un campo de dirección y un campo de datos. El sistema operativo en disco da acceso a los sectores individuales de una pista valiéndose del campo de dirección, que contiene los números de sector y de pista, y de un identificador (para verificar si el usuario está leyendo el disco correcto). Por lo tanto, puede recuperar información de modo bastante similar al de su recuperación de una dirección de memoria (utilizando su dirección)



Placa impresa analógica
Este sistema de circuitos convierte las señales que salen de la cabeza o llegan a ella. Traduce la forma digital utilizada por la máquina en la forma analógica que requiere el disco

Indicador
Este LED (diodo emisor de luz) indica si se le está dando acceso a la unidad de disco

Eje de transmisión
Engancha el disco plástico y lo hace girar en el interior de su contenedor

El ordenador personal "olvidará" toda la información con la cual usted lo había programado tan pronto como se interrumpa la alimentación eléctrica. Este hecho puede implicar, en el mejor de los casos, una pequeña molestia, y, en el peor, un gran disgusto: ver cómo ha desaparecido inadvertidamente una programación cuya realización le ocupó toda una tarde. Por esta causa los fabricantes de ordenadores personales incorporan un procedimiento que permite almacenar con carácter permanente el contenido de la memoria del ordenador. Éste consiste, por lo general, en una cinta de cassette en la cual el programa se almacena digitalmente como una serie de tonos (véase p. 94).

Sin embargo, cuando se trata de programas extensos o de una serie de programas cortos que han de utilizarse con frecuencia, el tiempo que se requiere para localizar y cargar el programa desde una cassette puede significar un serio contratiempo. Y ello se debe a dos razones. La primera es que para poder localizar un programa grabado en cinta, ésta se debe reproducir desde el principio (para lo que son de gran ayuda las grabadoras con contador de vueltas).

La segunda razón a la que aludíamos está relacionada con la forma en que se almacena el programa. Los patrones de bits retenidos en la memoria han de convertirse en una secuencia de tonos correspondiente: un tono alto representa a un bit que está encendido (o fijado en uno) y un tono más bajo representa a un bit que está apagado (o fijado en cero). Estos tonos deben luego grabarse en la cinta de la cassette. En la práctica, la mayor velocidad a la cual puede realizarse esta transferencia es de 150 bytes por segundo. Si se supera esta velocidad, el margen de error se incrementa hasta tal punto que el sistema deja de ser fiable.

Un sistema de cassette convencional que utilice una cinta C-10 puede invertir hasta cinco minutos por cada cara para hallar y direccionar un programa, siempre que se emplee un sistema de carga rápido. Algunos sistemas funcionan a una velocidad de tan sólo 30 bytes por segundo. Para estos programas extensos se necesitaría un sistema que hallara el comienzo del programa y lo cargara en cuestión de segundos.

Un sistema de almacenamiento de estas características es el disco flexible (conocido también como *floppy* o *diskette*), cuya utilización es viable en la mayoría de los ordenadores personales actuales. Si imagina los metros de cinta almacenados en una cinta de cassette que adopta la forma de un disco giratorio de aproximadamente cinco pulgadas de ancho (12,70 cm), comprobará con qué rapidez se puede localizar cualquier información almacenada en el disco. Éste viene alojado dentro de un contenedor protector y se introduce en una unidad de disco. Ésta tiene la función de hacer girar el disco (en el interior de su contenedor) a una velocidad constante, así como de proveer los medios para transferir los programas del disco al ordenador y viceversa. Esto lo realiza a través de una cabeza de grabación y reproducción, similar a la de las grabadoras de cassette aunque mucho menor. A diferencia de la de la cassette, que sólo pasa la cinta hacia adelante, esta cabeza puede moverse adelante y atrás a través de la superficie del disco mientras éste gira.







¡NO LO DOBLE! ¡NO LO APILE! APÁRTELO DE CAMPOS MAGNÉTICOS ALMACÉNELO CUIDADOSAMENTE MANTÉNGALO A TEMPERATURA AMBIENTE

El cuidado de los diskettes
Los discos flexibles son delicados y deben manejarse con gran cuidado. Atienda cuidadosamente las recomendaciones del fabricante

Cable plano
La información se transmite a, y desde, la unidad de disco a través de este cable plano. Contiene ocho bits y otras señales de control

Conexión del cable plano
Proporciona una conexión segura, aunque desmontable, del cable plano

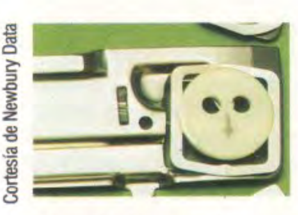
Engranaje motor
Hace girar el eje de transmisión

Cabeza de lectura-escritura

Motor de transmisión
Motor de gran precisión que hace mover la cabeza a través de la superficie del disco

Mecanismo de sujeción del disco
Conectado con el alerón de la puerta, este mecanismo de palanca proporciona al disco una situación exacta en el eje de transmisión

Cortesía de Newbury Data



Cabeza de lectura-escritura
Esta fotografía es una gran ampliación de la cabeza que lee y escribe la información en la superficie del disco. Es parecida a la cabeza de una grabadora de cassette, pero para el ojo humano es casi invisible

A diferencia de una cinta, que no es más que una larga cadena de bytes, un disco se compone de una serie de círculos concéntricos, cada uno de los cuales es tratado como trozos pequeños, por lo general de 256 bytes cada uno. Cada "sector" tiene una dirección.

Cuando se ha de escribir un programa en el disco, lo primero que sucede es que la cabeza se dirige hacia el directorio, un archivo especial que actúa a modo de índice de todo el disco. La cabeza examina el directorio para decidir dónde coloca el archivo. Si el programa se está reescribiendo, encuentra el primer sector de la copia antigua y la nueva información se almacena a partir de allí. Si se trata de un archivo nuevo, éste no tendrá entrada propia en el directorio, por lo cual habrá de otorgársele una: en este caso, la información se localiza en el primer sector libre, llenándose, de ser necesario, más sectores.

Las ventajas que ofrece el disco gracias a su expedito rendimiento y a su gran capacidad de almacenamiento explican la sustancial diferencia de precio entre una unidad de disco y una grabadora. La primera podría costar hasta seis veces más que una unidad de cassette. Esta diferencia de precio refleja el alto nivel tecnológico que requiere la producción de unidades de disco. La cabeza de grabación y reproducción de ésta es casi invisible y ha de colocarse en un espacio que mide apenas unas centésimas de pulgada.

El mecanismo que hace mover a esta minúscula cabeza se basa en un motor eléctrico que puede girar en fracciones de un grado. Está acoplado a un eje que transporta a la cabeza y la desliza a través de la superficie del disco en pasos calculados con suma precisión. Para asegurar que el disco gira a una velocidad constante se utilizan complejos sistemas electrónicos, y todos los componentes están montados sobre una estructura troquelada que se caracteriza por su especial resistencia, al objeto de reducir los efectos del calor y las vibraciones.

Desafiando a los elementos

A diferencia de sus congéneres simples, las variables con subíndice pueden contener cualquier número de elementos

En nuestro programa anterior para calcular cuántos días faltaban para la Navidad, nos encontramos con un nuevo tipo de variable denominada *variable subíndice*. Ésta se diferencia de las variables corrientes o "simples" en que en el interior de la caja puede albergar un número cualquiera de elementos. Las variables simples reconocen dos letras o letras seguidas de un dígito entre 0 y 9 (algunas versiones de BASIC permiten utilizar palabras enteras para los nombres de las variables). A, B, B1, C3 y R2 son todas variables simples. Las variables subíndice son, por ejemplo, de esta manera: A(6), B(12) o X(20). El número entre paréntesis es el subíndice. Los ejemplos que hemos dado se leen así: "A sub seis", "B sub doce" y "X sub veinte".

Si imaginamos a una variable simple como una caja que posee un nombre o una etiqueta, podemos asimismo pensar que una variable subíndice es una caja que contiene una cantidad específica de elementos internos. Si deseamos una variable con doce elementos, la creamos utilizando en primer lugar la sentencia DIM, de esta manera: DIM A(12). Se puede emplear cualquier letra del alfabeto.

A las variables simples se les asigna un valor de forma directa, utilizando tanto la sentencia LET como la sentencia INPUT, como: LET A=35, LET B1=365 o INPUT C3. A los elementos de las variables subíndice se les asigna valores de la misma manera. Veamos ahora cómo podemos asignarle valores a una matriz subíndice. (*Matriz* es otra palabra para designar un conjunto de variables subíndice.) Por ejemplo:

```
10 DIM A(5)
```

crea una variable subíndice con cinco elementos. Ahora podemos asignarle un valor a cada uno de estos elementos:

```
20 LET A(1)=5
30 LET A(2)=10
40 LET A(3)=15
50 LET A(4)=20
60 LET A(5)=100
```

Para comprender la diferencia que existe entre estas variables y las variables simples, asignemos valores a algunas variables simples:

```
70 LET X=5
80 LET Y=6
90 LET Z=7
```

Dé entrada a estas variables en su ordenador y luego verifique el contenido de cada una utilizando la orden PRINT. En BASIC muchas de las sentencias funcionan también como órdenes. Después de que haya dado entrada a las sentencias anteriores, verifíquelas pulsando LIST, pero después no digite RUN. En vez de RUN, digite PRINT X<CR>. En la pantalla debería visualizarse 5 instantáneamente. A continuación digite PRINT Y. El ordenador responderá a esta orden PRINT visualizando 6 en la pantalla. Si desea verificar los elementos de la variable subíndice, digite PRINT A(1) para averiguar el valor del primer elemento de la matriz. El ordenador debe responder imprimiendo 5 en la pantalla. Intente imprimir (PRINT) los valores de A(3) y A(5).

La importante diferencia entre las variables subíndice y las variables corrientes radica en el hecho de que el subíndice puede ser una variable en sí mismo. Para comprender lo que esto significa, digite PRINT A(X). La pantalla responderá con el número 100. ¿Por qué?

Observe la lista que ha digitado y luego verifique el valor de la variable X. Es 5. A(X) equivale a A (el valor de la variable X) y ésta equivale a A(5). Digitar PRINT A(X) es, por lo tanto, exactamente lo mismo que digitar PRINT A(5). ¿Qué valor esperaría si digitara PRINT A(Y−X)? Antes de intentarlo, trate de calcular usted mismo el resultado.

Asignando valores

Si sólo hay unas pocas variables simples, el modo más sencillo de asignarles valores es mediante la sentencia LET. Pero las variables subíndice bien pueden poseer en la matriz una gran cantidad de elementos; veamos, entonces, cuáles son los procedimientos alternativos para dar entrada a sus valores:

```
10 DIM A(5)
20 PRINT "ENTRE LAS VARIABLES"
30 INPUT A(1)
40 INPUT A(2)
50 INPUT A(3)
60 INPUT A(4)
70 INPUT A(5)
```

La digitación de este procedimiento es tan tediosa como si se emplearan sentencias LET, aunque funcionará con seguridad. Si sabemos exactamente cuántas variables hay (en este caso hay cinco), es más fácil utilizar un bucle FOR-NEXT como éste:

```
10 DIM A(5)
20 FOR X=1 TO 5
30 INPUT A(X)
40 NEXT X
```

Según este programa, para ejecutarlo se habrían de digitar cinco valores en el teclado del ordenador. Después de dar entrada a cada número habría que pulsar la tecla RETURN. Si sabemos de antemano cuáles son los valores de la variable, es mucho más sencillo darles entrada utilizando una sentencia READ junto con una sentencia DATA, de este modo:

```
10 DIM A(5)
20 FOR X=1 TO 5
30 READ A(X)
40 NEXT X
50 DATA 5, 10, 15, 20, 100
```

Pruebe con este corto programa y luego verifique los contenidos de la matriz utilizando la orden PRINT; es decir, utilice PRINT después de haber ejecutado (RUN) el programa. Por ejemplo, PRINT A(1) <CR> y PRINT A(5). Ahora podemos agregarle algunas líneas al programa para que se nos impriman automáticamente los elementos de la matriz:

```
60 FOR L=1 TO 5
70 PRINT A(L)
80 NEXT L
90 END
```

Ejecute (RUN) este programa y compruebe que en la pantalla se impriman los valores correctos. Después vuelva a digitar la línea 50 utilizando cinco ítems de DATA diferentes. Recuerde que en una sentencia DATA los números han de ir separados entre sí mediante comas, pero que no debe haber una coma antes del primer número ni después del último.

La forma más sencilla de asignar valores es utilizando las sentencias READ y DATA. Si los valores van a ser diferentes cada vez que se ejecute el programa, probablemente lo mejor sea emplear la sentencia INPUT dentro de un bucle FOR-NEXT. En el caso de que el número total de elementos en la matriz sea fijo, este número puede ser utilizado como límite máximo en la sentencia FOR.

Apliquemos todo cuanto llevamos aprendido hasta ahora para construir un programa corto pero eficaz. Supongamos que deseamos clasificar algunos números por orden ascendente. Antes de ponernos a escribir el programa, lo primero que debemos hacer es hallar una forma lógica de resolver el problema. Cuando la solución a éste parezca clara, habremos de escribir uno por uno todos los pasos utilizando oraciones cortas y concisas.

Supongamos que empezamos con cinco números: 4, 9, 2, 8, 3. Clasificarlos por orden ascendente es una tarea irrelevante. Simplemente, miramos la línea para ver cuál es el más pequeño y lo colocamos a la izquierda; luego repetimos el mismo proceso para los dígitos restantes.

El ordenador, sin embargo, necesita de una serie de instrucciones muy precisas, de modo que nosotros hemos de pensar con mucha claridad los pasos que se requieren. Un enfoque es el siguiente: compare el primer dígito con el segundo dígito. Si el primer dígito es mayor que el segundo deje el primero y coja el segundo. Si el primer dígito es menor que el segundo, no los cambie de lugar.

Compare el segundo dígito con el tercer dígito. Si el segundo dígito es menor que el tercero, déjelos en la misma posición.

Repita el proceso de comparar los dígitos de a pares hasta haber comparado el último par.

Si no hubo que cambiar ninguna posición, todos los números han de estar en orden. Si, por el contrario, hubo que invertir la posición de algún par, vuelva al comienzo y repita la operación.

Si analiza este proceso comprobará que, efectivamente, sirve para clasificar cualquier serie de números según un orden ascendente. Observe lo que sucedería con nuestra serie original de números a medida que los dígitos se comparan de a pares:

```
4 9 2 8 3
4 2 9 8 3
4 2 8 9 3
4 2 8 3 9
```

Ahora todos los pares han sido comparados y, cuando ha sido necesario, se han invertido las posiciones. Puesto que se ha producido por lo menos una inversión de posiciones, volvamos a comenzar y repitamos el proceso:

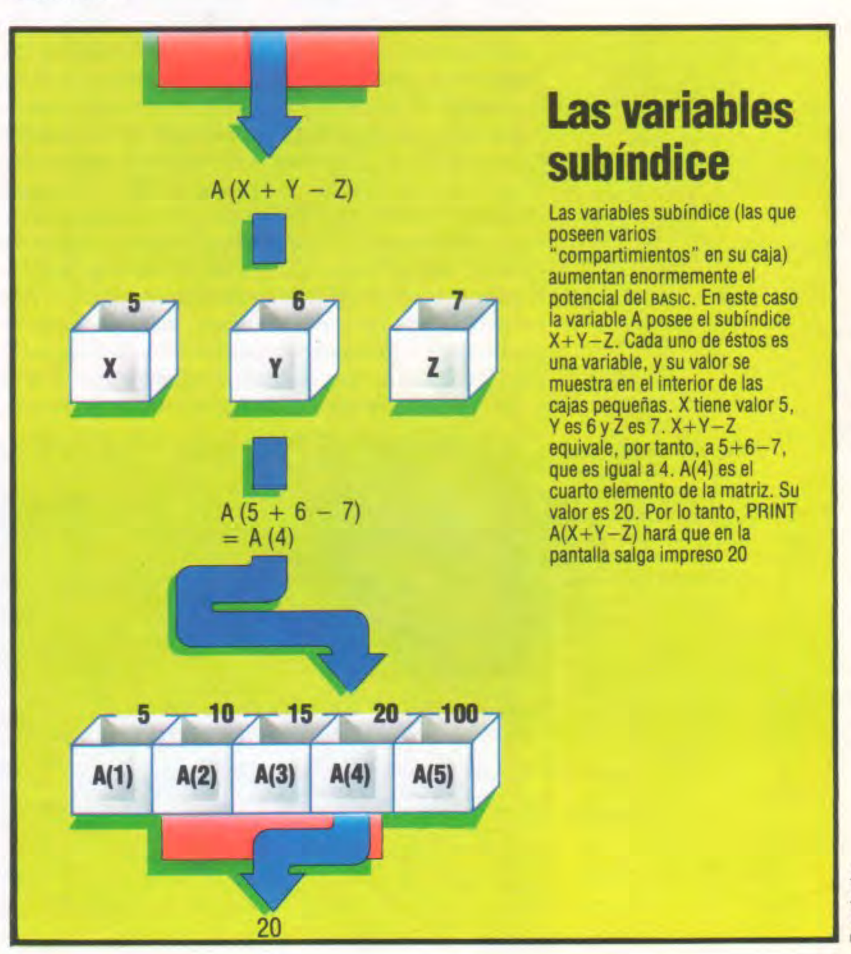
```
4 2 8 3 9
2 4 8 3 9
2 4 3 8 9
2 4 3 8 9
```

Aún se han producido dos inversiones, por lo tanto volvamos al principio y repitamos:

```
2 4 3 8 9
2 3 4 8 9
2 3 4 8 9
```

En esta última no ha habido inversiones, de modo que todos los números han de ser menores que el número situado a su derecha. Los números deben estar por orden ascendente y, en consecuencia, puede terminarse la operación.

La utilización de las variables subíndice permite que en BASIC una rutina de clasificación como ésta se realice fácilmente, porque el subíndice puede ser, en sí mismo, una variable. Si nuestros cinco números originales fueran los valores de una matriz, como: A(1)=4, A(2)=9, A(3)=2, A(4)=8 y A(5)=3, y si el valor de X fuese 1, luego A(X) sería el contenido de A(1), que es 4. A(X+1) sería el contenido de A(2), que es 9, y así sucesivamente.



Las variables subíndice

Las variables subíndice (las que poseen varios "compartimientos" en su caja) aumentan enormemente el potencial del BASIC. En este caso la variable A posee el subíndice X+Y−Z. Cada uno de éstos es una variable, y su valor se muestra en el interior de las cajas pequeñas. X tiene valor 5, Y es 6 y Z es 7. X+Y−Z equivale, por tanto, a 5+6−7, que es igual a 4. A(4) es el cuarto elemento de la matriz. Su valor es 20. Por lo tanto, PRINT A(X+Y−Z) hará que en la pantalla salga impreso 20

Tony Lodge

Observe el programa y vea si comprende exactamente lo que está sucediendo. La línea 20 establece que el valor de la variable N es la cantidad de números que deseamos clasificar. Supongamos que deseamos clasificar cinco números: al ejecutar el programa digitaremos 5 y luego pulsaremos RETURN.

La línea 30 es la sentencia DIM (DIMensión). Si N es 5, establece en 5 el tamaño de la matriz. Esta línea equivale a DIM A(5).

Entre las líneas 40 y 60 hay un bucle FOR-NEXT que nos permite digitar los cinco números. La mayoría de las versiones de BASIC alertan al usuario con un signo de interrogación en la pantalla. Después de dar entrada a cada uno de los números se debe pulsar RETURN. Los números pueden ser de más de un dígito y pueden incluir fracciones decimales.

La línea 90 establece que la variable S es 0. Esta variable se está utilizando como "señal". Más adelante en el programa, A se compara para ver si es o no es 1. Sólo se establece en 1 en el supuesto de que se haya invertido la posición de dos números, como veremos más adelante, en la línea 240. Más avanzada nuestra obra estudiaremos con mayor detalle de qué manera se utilizan las "señales".

La línea 100 establece los límites de un bucle; en este caso, entre 1 y 4 (porque, como N es 5, N − 1 es 4). La primera vez que se realiza el bucle L es 1, de modo que A(L) en la línea 110 será A(1) o el primer elemento de la matriz, y A(L+1) será A(2), el segundo elemento de la matriz. La próxima vez que se realice el bucle, L se aumentará a 2, de modo que A(L) equivaldrá a A(2) y A (L+1) equivaldrá a A(3). La línea 110 establece una comparación para ver si A(L) es mayor que el número de la matriz que se encuentra situado inmediatamente a su derecha. El signo "mayor que" es >.

Si el primer número es mayor que el siguiente, el programa se bifurca hasta una subrutina que invierte la posición de los números. Si el primer número no es mayor que el siguiente, no se produce la bifurcación hacia la subrutina y el BASIC simplemente continúa con la siguiente línea, que es la sentencia NEXT L. Después de que el bucle se ha repetido cuatro veces, el programa se detiene y va hasta la línea 130 que compara la señal S, "de inversión", para comprobar si se ha establecido o no. Si se ha establecido (en la subrutina "de inversión"), el programa se bifurca hasta la línea 90 para repetir el proceso de comparación. Si S no es 1, ello significa que no se ha producido ninguna inversión y que, por lo tanto, todos los números están por orden. El resto del programa es tan sólo para imprimirlos.

Complementos al BASIC



Si este programa es para ejecutarse en el Spectrum, la línea 130 ha de modificarse para que diga: 130 IF S=1 THEN GOTO 90



El Spectrum no dispone de esta sentencia y, por lo tanto, se debe cambiar la línea 170 para que quede: 170 GOTO 260, y se debe agregar la línea 260 REM FIN DEL PROGRAMA



Este programa no funcionará en el TI 99/4A, debido a que los subíndices, como X en la línea 50 y L en la línea 110, han de ser números específicos y no variables

Para poder almacenar uno de los números que han de invertirse, la subrutina "de inversión" ha de ser una variable temporal. Después de que en las líneas 210, 220 y 230 los dos números se han invertido, la señal "de inversión" S se establece en 1 y entonces el programa retorna (RETURN) al programa principal.

```
10 PRINT "¿CUANTOS NUMEROS DESEA
   CLASIFICAR?"
20 INPUT N
30 DIM A(N)
40 FOR X=1 TO N
45 PRINT "NUMERO SIGUIENTE"
50 INPUT A(X)
60 NEXT X
70 REM
80 REM RUTINA DE CLASIFICACION
90 LET S=0
100 FOR L=1 TO N-1
110 IF A(L) > A(L+1) THEN GOSUB 200
120 NEXT L
130 IF S=1 THEN 90
140 FOR X=1 TO N
150 PRINT "A(";X;")=";A(X)
160 NEXT X
170 END
180 REM
190 REM
200 REM SUBRUTINA DE INVERSION
210 LET T=A(L)
220 LET A(L)=A(L+1)
230 LET A(L+1)=T
240 LET S=1
250 RETURN
```

Ejercicios

■ Extienda el programa para hallar el valor promedio de la entrada de números. El promedio es igual a la suma de los ítems dividido por el número total de ítems. La forma más sencilla de hacerlo consiste en insertar un GOSUB justo antes de la sentencia END de la línea 170. La subrutina habrá de leer cada uno de los elementos de la matriz y agregar los valores para dar una variable "suma". Después de que se hayan sumado todos los elementos, la suma habría de dividirse por el número total de los elementos. La suma se deriva con bastante mayor facilidad utilizando el número de elementos como límite máximo de un bucle FOR-NEXT.

■ Modifique una línea del programa de manera tal que los números queden clasificados según un orden descendente.

■ Este ejercicio está dirigido especialmente a los usuarios del TI99/4A, ordenador que no admite que se utilicen variables como subíndices de variables subíndice. No obstante, el BASIC de TI acepta sentencias similares a DIM A(12). Reescriba el programa de modo que la sentencia INPUT espere la entrada de una cantidad exacta de números, por ejemplo, 12. Esto evitará el problema de tener que utilizar como subíndice un nombre de variable. También habrán de modificarse las líneas 100 y 110. Por la misma razón, la subrutina de inversión no funcionará en el BASIC del TI. También esto tendrá que modificarse.

■ He aquí un ejercicio difícil. El modo de clasificar los números que hemos elaborado no es, de ninguna manera, el único que existe. Intente pensar un procedimiento alternativo.

Multiplicación

Los ordenadores dan respuestas muy rápidas a complejos problemas aritméticos, abordándolos del modo más sencillo

En la última parte de nuestro curso acerca del sistema binario, descubrimos cómo suman los ordenadores. A continuación estudiaremos el proceso de la multiplicación.

Si tuviera que multiplicar 14 por 12, una forma sencilla de hacerlo sería mediante una suma múltiple, por ejemplo 14+14+14+14+... (12 veces). Dado que, en cierto sentido, la multiplicación es una suma repetida, con toda seguridad este método funcionará. (Ésta era, precisamente, la forma en que multiplicaban los primeros ordenadores.)

Sin embargo, este procedimiento es tosco y ocupa mucho tiempo, de modo que los diseñadores de estas maravillosas máquinas tuvieron que desarrollar un método más eficaz.

Cuando se multiplican dos números, normalmente la operación se escribe así:

```
  14
 ×12
 ---
  28
 +14     (a menudo se suele poner
 ---      un 0 final para ayudar a colocar
 168      los dígitos en la columna correcta)
```

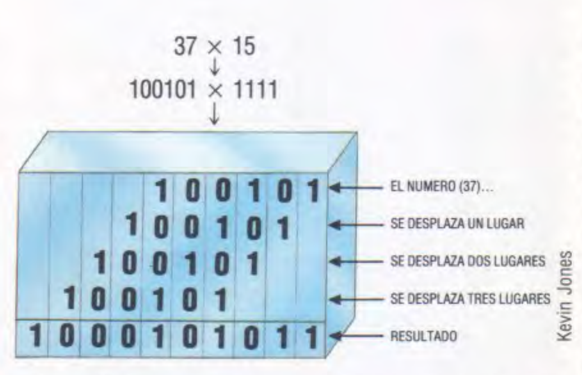
El proceso funciona exactamente de la misma manera con cualquier base de números. Tomemos un ejemplo binario o de base dos:

```
  101
  ×11
 ----
  101
+101
 ----
 1111
```

Aun con números más elevados el procedimiento es exactamente el mismo que hemos expuesto, de modo que volvamos al ejemplo de 14×12 y resolvámoslo en binario:

```
     1110    (14)
    ×1100    (12)
 --------
     0000
    0000
   1110
  1110
 --------
 10101000    (168)
```

En el sistema binario la multiplicación es todavía más sencilla que en el decimal, porque nunca puede llevarse un dígito. Cuando se multiplica un número por 1, dicho número no se modifica: 14 × 1 = 14; y cuando se multiplica un número por 0, el resultado es sencillamente 0: 14 × 0 = 0. Y esto ocurre así tanto en binario como en decimal y en cualquier otro de los sistemas numéricos.

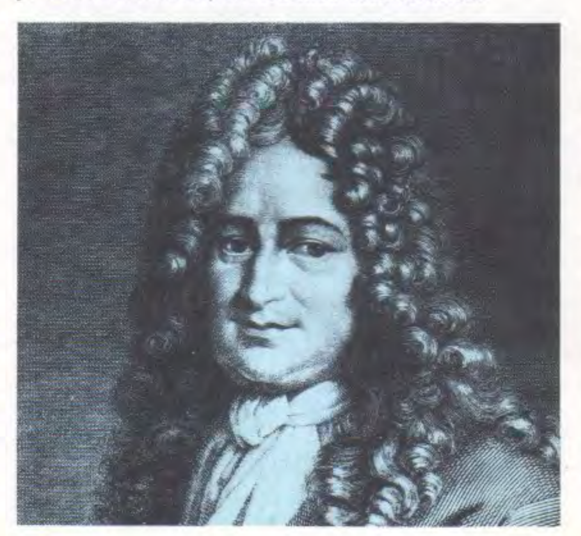


El desplazamiento en la multiplicación
Multiplicar en binario es mucho más fácil que multiplicar en decimal. Se sigue el mismo proceso de multiplicación larga que se aplica en el sistema decimal; pero como sólo hay dos números implicados en la multiplicación (0 y 1), la operación es muy sencilla. Cuando un número se multiplica por 1 el resultado es, obviamente, el mismo número. En la ilustración, donde se multiplica 100101 (37) por 1111 (15), aparecen cuatro copias de 100101. Cada copia se desplaza hacia la izquierda de acuerdo a la posición del 1 por el cual se multiplique. Finalmente, se suman todas las copias y se obtiene el resultado de 1000101011 (555)

Cuando los matemáticos estudiaron cálculos similares al que mostramos arriba, vieron que se producía un patrón sencillo: la multiplicación en binario consiste en sólo dos operaciones: desplazamiento y suma. Y así es exactamente como multiplica el ordenador. Primero desplaza "copias" de la línea superior a su posición correcta (determinada por los unos y los ceros de la línea inferior) y luego suma todas las "copias" juntas.

Para poder multiplicar, el ordenador necesita una gran capacidad de dígitos. Cuando el número de 4 dígitos 1110 multiplicó por el número de 4 dígitos 1100, el resultado fue de 8 dígitos (10101000) y, en general, el resultado de una multiplicación puede tener una longitud que llegue a equivaler hasta el doble de la del número mayor.

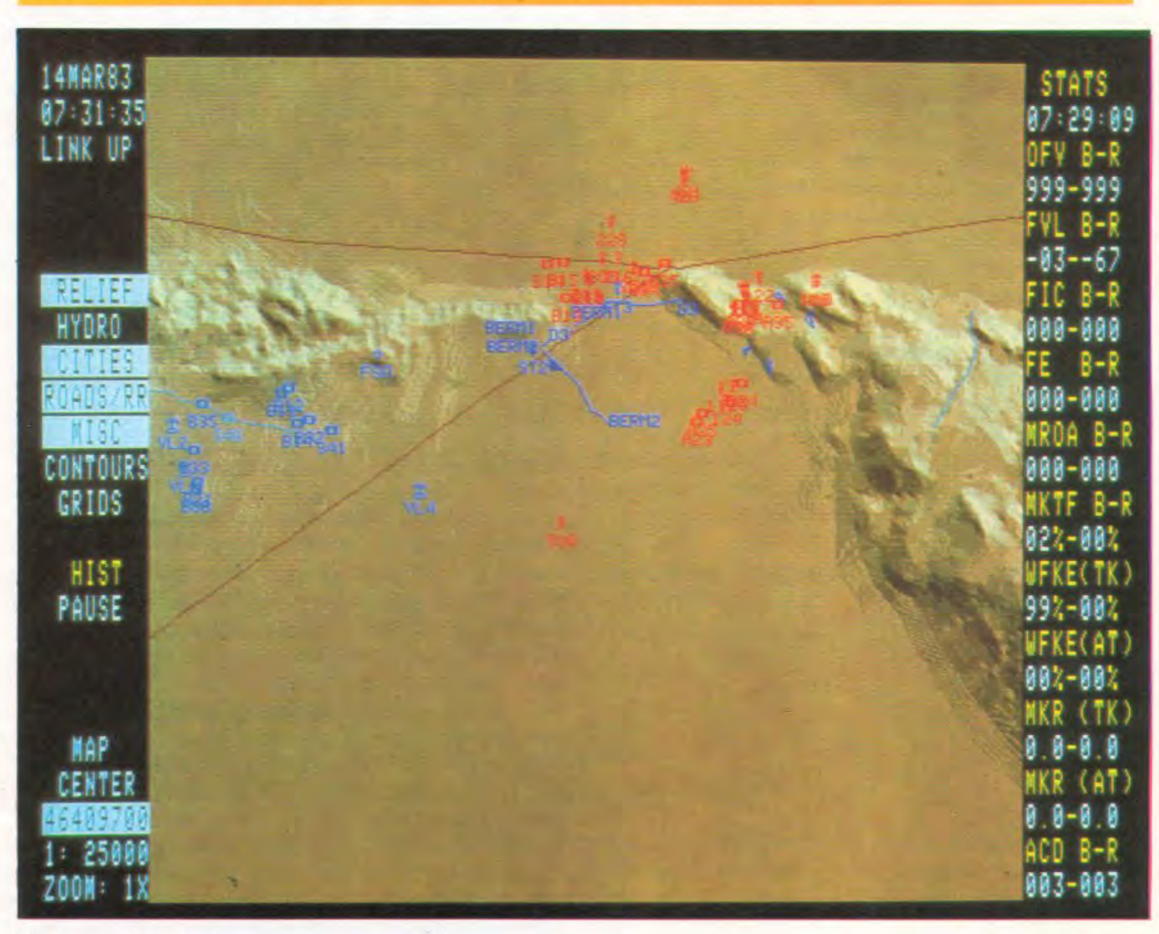
Puede constituir una sorpresa enterarse de que los ordenadores pueden equivocarse al multiplicar. Prácticamente todos estos errores se producen en función de la cantidad de espacio que el diseñador de la máquina haya previsto para retener el resultado. Si no existe espacio suficiente se producirá un "desbordamiento", se perderán los dígitos menos significativos y, en consecuencia, el resultado será erróneo.



Gottfried Wilhelm Leibniz
Leibniz (1646-1716) fue contemporáneo de Isaac Newton e hizo aportaciones en el campo de las matemáticas, la ciencia y la filosofía. Inventó una máquina que multiplicaba y dividía. También investigó las posibilidades de utilizar la aritmética binaria en aparatos de cálculo, si bien el primer antecedente conocido en cuanto a números binarios se atribuye a Francis Bacon, y data de 1623. En los últimos años de vida entabló una polémica con Newton acerca de la invención del cálculo

Micros al mando

La aplicación del ordenador en la tecnología militar ha sido un factor decisivo en el desarrollo de la informática



En el campo de batalla
El ejército norteamericano en un simulacro en el desierto de Mojave. Las ametralladoras y otras armas están conectadas con sistemas láser y sus objetivos tienen dispositivos sensores. Cuando el rayo láser golpea un sensor, el ordenador graba la precisión del impacto





Brian Wolff

En la actualidad tal vez las fuerzas armadas sean las "industrias" más computerizadas del mundo. El soldado moderno, además de estar bien entrenado para el combate tradicional y para desarrollar una gran capacidad de resistencia, debe ser asimismo capaz de operar la tecnología que emplean las armas modernas.

La guerra de las Malvinas demostró el poder devastador que posee la nueva generación de armas por microchip. El misil Exocet que hundió al *Sheffield* británico estaba guiado por ordenador y el piloto ni siquiera necesitó mirar el objetivo y apuntarle en la forma convencional.

Los ordenadores están demostrando ser enormemente útiles para los tanques, en los que la precisión y la velocidad son factores esenciales. Los carros de combate Chieftain, del Ejército británico, están equipados con un ordenador que controla factores de importancia primordial, como son la dirección del viento, la posición del tubo del cañón, la clase de munición que se está utilizando y el ángulo de tiro. A partir de esta información calcula la ruta correcta y apunta hacia el blanco con precisión. Un ordenador de este tipo le permite al tanque acertar nueve impactos de nueve disparos, en un lapso de 53 segundos y con un alcance de 1,2 km.

Los ejércitos de la OTAN están utilizando un ordenador similar. Se trata de un sistema belga que emplea un telémetro láser, sensores, un ordenador y un visor óptico. Los sensores miden factores como el viento y la posición del cañón, y el ordenador calcula los ángulos y luego expone en la mira una serie de hilos del retículo. Como los hilos siguen al blanco, lo mismo hace el cañón.

Los misiles Cruise emplean diversos ordenadores para apuntar su cabeza explosiva a una distancia de 18 m del blanco, después de un vuelo de aproximadamente 3 200 km. Cuando se lanza el misil, su vuelo es guiado por un ordenador que almacena toda la información relativa a la ruta seguida. El ordenador realiza de manera continua ligeros ajustes en la trayectoria de vuelo. Cuando el misil se está aproximando al objetivo, el ordenador activa su sistema de guía final e identifica el objetivo a partir de una "imagen" compuesta por millones de números que tiene en su memoria. Una vez los detectores del misil han "visto" el objetivo, le envían al ordenador otra imagen digital. Entonces el ordenador guía al misil hasta que las dos series de números coinciden exactamente, momento en que lo dirige hacia el blanco.

El desarrollo de los ordenadores para su aplicación a la tecnología militar ha sido un factor primordial para acelerar el progreso de la ciencia y el diseño de la informática. Sin las enormes sumas de dinero que se invirtieron en los laboratorios militares, es muy probable que jamás en toda nuestra vida hubiésemos entrado en contacto con un ordenador personal.

Microingeniería y microelectrónica

Los científicos descubrieron que el microchip podía desarrollarse utilizando uno de los elementos más abundantes en la naturaleza: el silicio

El silicio existe en estado natural en toda la superficie de este planeta con mayor abundancia que cualquier otro elemento excepto el oxígeno, con el cual se combina para formar la sílice. Muchos de nosotros pasamos las vacaciones estivales tumbados sobre ella. ¡Toda la revolución de la microelectrónica se basa en la arena!

La importancia del silicio para la industria de la microelectrónica reside en su estructura física. En su forma pura, el silicio es un pésimo conductor de la electricidad. Sin embargo, cuando se introducen cantidades determinadas de ciertas impurezas, se convierte en un semiconductor.

La conducción de la electricidad a través de una sustancia viene determinada por el número de electrones de cada uno de sus átomos y por el tipo de enlace. En los metales, la corriente eléctrica es transmitida por los electrones libres. Éstos tienen un movimiento desordenado en el interior de la estructura atómica y transmiten su carga eléctrica de un átomo a otro. En un aislante, todos los electrones están enlazados firmemente; por tanto, la corriente no puede pasar de un punto a otro.

La obtención de silicio puro es un proceso sencillo. Primero se refina químicamente el óxido de silicio natural hasta obtener una pureza del 99,99 %. Luego se introduce en un crisol y se calienta hasta que alcance su punto de fusión (1 410 °C) en una atmósfera de gas noble purificado, para evitar la presencia de sustancias extrañas.

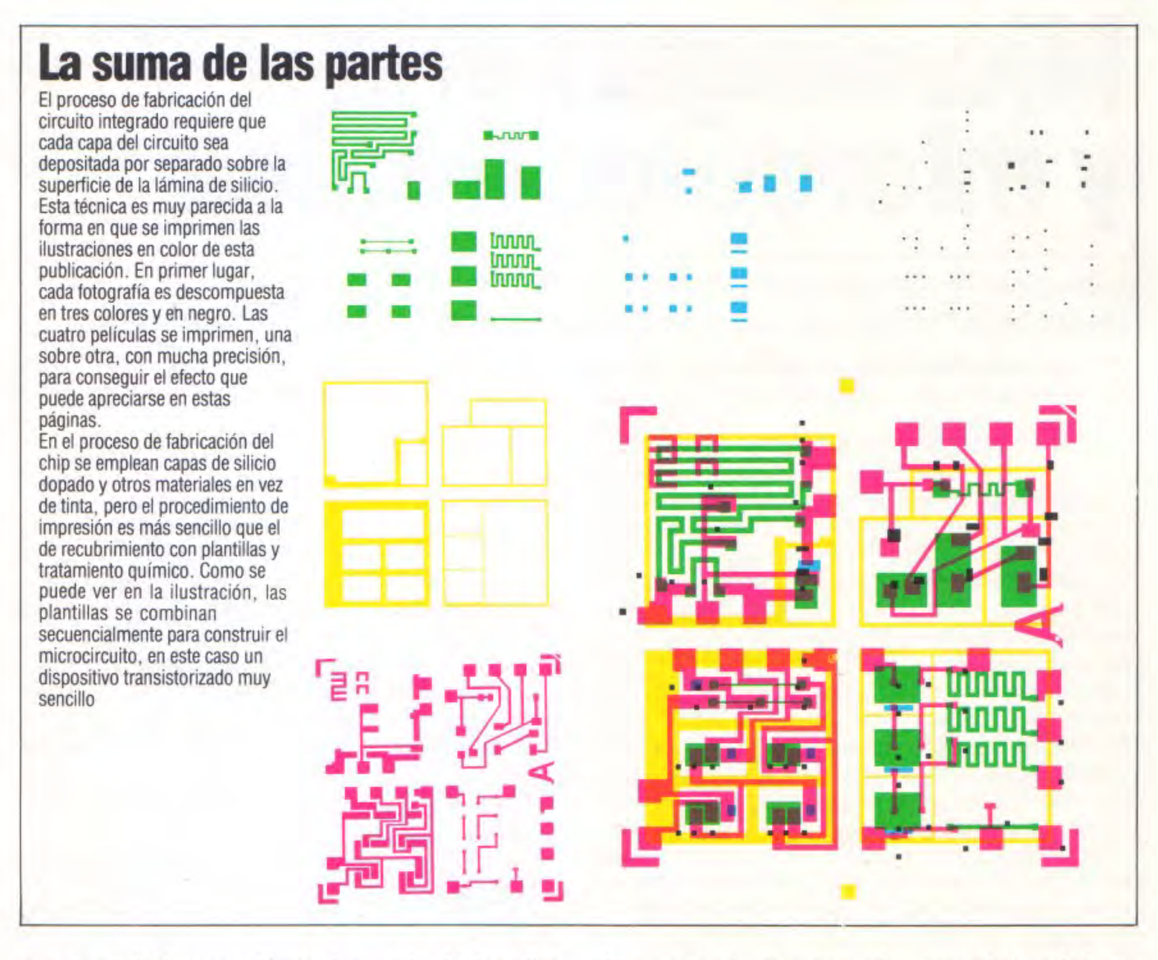
El proceso de introducción de cantidades determinadas de impurezas (conocido como *dopado*) hace que el silicio puro se combine con el fósforo, obteniéndose silicio del "tipo n" (así llamado por transportar la



Cortesía de Ferrant

El horno de fabricación de chips
La fotografía muestra un horno de chips en funcionamiento. Una vez que el silicio ha sido refinado, cortado en láminas, pulido, cubierto con las plantillas y tratado químicamente, se le debe dar una capa de dióxido de silicio. Para ello, se calientan las láminas a 1 050 °C y se las somete a una corriente de oxígeno muy puro o de vapor de agua sobrecalentado. Mientras las láminas permanecen en el horno, que está hecho de cuarzo fundido, sobre ellas se forma una capa de dióxido de silicio. Esta capa es extraída selectivamente por el proceso químico a que son sometidas a continuación. El ciclo se repite para cada capa del chip





La suma de las partes

El proceso de fabricación del circuito integrado requiere que cada capa del circuito sea depositada por separado sobre la superficie de la lámina de silicio. Esta técnica es muy parecida a la forma en que se imprimen las ilustraciones en color de esta publicación. En primer lugar, cada fotografía es descompuesta en tres colores y en negro. Las cuatro películas se imprimen, una sobre otra, con mucha precisión, para conseguir el efecto que puede apreciarse en estas páginas.
En el proceso de fabricación del chip se emplean capas de silicio dopado y otros materiales en vez de tinta, pero el procedimiento de impresión es más sencillo que el de recubrimiento con plantillas y tratamiento químico. Como se puede ver en la ilustración, las plantillas se combinan secuencialmente para construir el microcircuito, en este caso un dispositivo transistorizado muy sencillo

carga negativa), o con el boro, formando el "tipo p", o silicio positivo.

Al introducir un "germen" perfecto de cristal en el silicio fundido y extraerlo lentamente, al mismo tiempo que se va girando, se obtiene un gran cristal. De esta forma se pueden obtener cristales de 7 a 10 cm de diámetro y 60 cm o más de longitud. Éstos son reducidos a un diámetro estándar de 76 o 100 mm. El cristal es cortado en láminas, que luego son pulidas por una sola cara. La "oblea" acabada suele tener un grosor de medio milímetro.

Si el proceso es razonablemente simple y la materia prima tan abundante en la naturaleza, ¿por qué es tan caro el silicio que se emplea en la fabricación de los chips?

La respuesta reside en la absoluta necesidad de mantener su pureza. Se debe tener un cuidado extremo en evitar la presencia de materias extrañas. Los niveles de pureza del aire en las plantas de fabricación son verdaderamente notables: menos de 3 000 partículas por metro cúbico. Es decir, una atmósfera cien veces más pura que la que se respira en los hospitales modernos.

La fabricación de un circuito integrado requiere un método de micrograbado de la superficie de los chips. En la producción en serie, esto se logra mediante un proceso llamado fotolitografía, similar en muchos aspectos a los métodos utilizados para realizar esta publicación.

Cada "capa" del circuito se trata durante el proceso de una forma totalmente independiente. El diseño original se realiza con ayuda de un ordenador, y se convierte en la fotografía de una línea o bloque, que es luego reducida a su tamaño real. La plantilla se forma reproduciendo esta fotografía numerosas veces, creando una especie de entramado, para cubrir toda la superficie de la lámina.

Primero se calienta la lámina a 1 050 °C en una atmósfera de oxígeno puro. Esto hace que se forme en la superficie una capa de dióxido de silicio, que actuará como aislante. Esta capa es luego extraída selectivamente para formar "ventanas" sobre el silicio puro. Este proceso se repite para cada etapa sucesiva de la fabricación del circuito integrado en la superficie del sustrato de silicio.

La superficie oxidada de la lámina se recubre con una capa de una sustancia "fotorresistente", es decir una sustancia sensible a la luz, cuya solubilidad se reduce en gran medida al exponerla a la radiación ultravioleta. Entre la superficie recubierta y la fuente luminosa se introduce una plantilla con la forma de la primera capa del microcircuito. La superficie es expuesta a la radiación ultravioleta y luego "revelada" en un disolvente que quita la sustancia fotorresistente no expuesta.

En la fabricación de grabados de cobre se utiliza un procedimiento similar. La superficie de una plancha de cobre se recubre con una capa de cera, en la que se marca un dibujo, y luego la plancha se sumerge en un ácido. Al quitar la cera, el dibujo permanece grabado en la superficie de cobre. En las zonas que han sido recubiertas con cera no se ha producido ninguna reacción química.

Una vez que se ha completado la primera etapa, se vuelve a repetir todo el proceso con diferentes plantillas y reactivos químicos hasta obtener el circuito deseado.

Por último, se recubre toda la lámina con otra capa de dióxido de silicio.

Este procedimiento puede requerir la repetición del

proceso completo unas diez o más veces. En cada una de las etapas, las posibilidades de error son grandes, y, por tanto, al efectuar el control de calidad y la verificación, el porcentaje de piezas desechadas es elevado. Cuanto más complejo sea el microcircuito y mayor el número de capas, mayor será la cantidad de rechazos que se produzca.

En primer lugar se verifica la lámina entera, con sus varios cientos de circuitos integrados idénticos. Esto se realiza con un equipo controlado por ordenador, que comprueba cada circuito.

Frecuentemente, en cada lámina se reservan nueve zonas del tamaño de un chip con fines de verificación y control de calidad.

El equipo de comprobación no sólo marca cada pieza desechada con un punto de tinta, sino que también extrae información sobre el porcentaje de fallos de cada lámina, localiza impurezas tanto en la lámina como en cada chip individual e identifica errores específicos.

A continuación, se cortan las láminas en chips individuales y las piezas desechadas se separan manualmente. Las restantes se montan sobre microestampaciones metálicas.

Las patillas de conexión del chip se engarzan en la estampación mediante hilos finísimos —proceso que es controlado otra vez por ordenador—. El conjunto es sellado con un soporte de plástico o cerámica, del que sobresalen las patillas de conexión. Finalmente se llevan a cabo nuevas y más exhaustivas comprobaciones.

En la actualidad, el límite en el grado de miniaturización, condicionado por el proceso fotolitográfico de fabricación, es la longitud de onda de la luz (alrededor de dos millonésimas de metro). Los avances recientes se centran en la utilización de rayos X, que probablemente permitirán reducir 50 veces el tamaño de los circuitos microelectrónicos.

Casi todas las etapas de la fabricación del chip, desde el diseño, pasando por los prototipos, hasta la fabricación en serie y la verificación final, son tan complejas que sería prácticamente imposible realizarlas sin la utilización de muchos de los elementos que a su vez están en proceso de fabricación. ¡Una paradoja notable!

El empleo de los ordenadores en el proceso de diseño, por ejemplo, permite que las subsecciones sean predefinidas y llamadas desde la memoria cada vez que se necesitan.

Tómese el caso de un chip RAM, en donde cada bit es mantenido en una celda de almacenaje, formada por un transistor sencillo. Para almacenar 2 Kbytes, el chip deberá contener nada menos que 16 384 celdas idénticas. Para el diseñador resulta sencillo definir la estructura de una "muestra" una sola vez y a continuación "enseñar" al ordenador de diseño a repetirla 16 384 veces.

El empleo de los ordenadores en el departamento de diseño no se limita a esta práctica y bastante fácil artimaña. Los proyectistas ahora no trabajan en tableros de dibujo, sino que "dibujan" directamente sobre la pantalla de un monitor. El acabado del dibujo es preparado directamente por el ordenador de diseño utilizando un dispositivo trazador de gráficos (llamado también *plotter*) multicolor.

No es únicamente el proceso de diseño gráfico lo que se ha mejorado con la mecanización. En la página 103 se describe la creación de un modelo mecanizado en el campo de la producción de aceite. Esta misma técnica puede aplicarse también al diseño de circuitos, permitiendo al diseñador probar diferentes soluciones antes de someterlo al costoso proceso de fabricación. Con este procedimiento puede evitarse buena parte de los gastos dedicados a la prueba de prototipos.

Cuando hablamos de "la revolución de la microelectrónica", nos estamos refiriendo a saltos cuantificados en la velocidad y a reducciones en el coste y en el tamaño. Quizá sea interesante en este momento recordar las principales características de un ordenador de válvulas:

gran tamaño;
lentitud de proceso;
alto consumo de potencia;
memoria limitada y poca capacidad de programación;
alto coste.

Todos estos factores se vieron afectados por el descubrimiento del transistor (véase p. 46), pero desde el punto de vista del proceso de fabricación, la industria del ordenador requería todavía numerosa mano de obra y, por tanto, era costosa. Los distintos componentes tenían que ser aún montados en paneles de circuitos impresos.

Fue precisamente el descubrimiento del circuito integrado —y en particular del microprocesador— lo que permitió que la industria sacara partido de los avances logrados en la fabricación y proceso controlados por ordenador.



Cortesía de Texas Instruments Ltd.

Un revolucionario de la microelectrónica
Generalmente, el descubrimiento del circuito integrado se atribuye a Jack Kilby, en 1958, cuando trabajaba para la compañía Texas Instruments. Kilby construyó un soporte de media pulgada por un cuarto que contenía varios transistores. Los circuitos electrónicos modernos, en el mismo espacio, contienen cientos de miles de dispositivos

Algunas de las estadísticas disponibles son impresionantes. Por ejemplo, en 1959 un circuito integrado sólo podía contener un componente: un diodo o un transistor. En 1978, el más denso de los circuitos integrados a gran escala (LSI) tenía más de un cuarto de millón de componentes en un simple chip. En un período aún más corto, de 1973 a 1983, el coste por bit de ordenador se ha reducido a la veinteava parte, y el uso de componentes electrónicos de cualquier tipo en todo el mundo se ha multiplicado por mil. Y esta tendencia es probable que continúe en el futuro. Se estima que el número de aparatos que se utilizan cada año se multiplicará por cien en el próximo trienio.

Selector

Un ordenador puede seleccionar hechos y recopilar listas a partir de la información almacenada en una base de datos

Una acumulación de datos almacenada en un ordenador y que sea accesible a éste, se conoce como "base de datos". Todos usamos diversas bases de datos no mecanizadas en nuestra vida cotidiana.

El listín de teléfonos es un ejemplo de base de datos no mecanizada. Sin embargo, la información no necesita estar clasificada o almacenada en un orden determinado para ser una base de datos. En un ordenador, ese orden determinado crea realmente limitaciones importantes.

Un programa de base de datos es una serie de rutinas que permite hacer una selección de datos. La gama de programas varía desde sistemas de fichas a lenguajes completos.

Normalmente, una base de datos mecanizada será amplia y contendrá información de muy diferente tipo. Pero ello no significa que sea necesario poseer una máquina de enormes dimensiones. Cualquier ordenador puede manejar una base de datos práctica. La única limitación real es el tamaño y la velocidad de la memoria.

Por ejemplo, podríamos confeccionar una lista denominada "gente" que contenga datos sobre varias personas. Si colocamos estos datos en fichas corrientes, obtendremos una lista similar a la denominada "índice personal". Resulta evidente que en ella hay diferentes clases de información. Dentro de cada categoría, muchos de los apartados son palabras simples, y otros son números. En dos de ellas las posibilidades son limitadas: "sexo" puede ser sólo hombre o mujer, y en "estado civil" deberá ser "soltero, casado, divorciado o viudo".

Podría ser útil colocar ciertos apartados en listas de palabras o números. Por ejemplo: profesión, nombre de la empresa, dirección profesional, teléfono profesional y el nombre del director podrían agruparse bajo el encabezamiento de "empresa", mientras que el modelo y número de años del coche pueden formar parte de la lista "coche".

Aplicando este mismo concepto, se puede hacer un listado con todas las direcciones. Esto es más práctico que mantener cada dirección por separado, ya que se puede desear saber en qué ciudad vive una persona, pero no en qué calle.

Asimismo también puede ampliarse el apartado "estado civil" añadiendo el nombre del cónyuge cuando sea aplicable. Éste puede consistir sólo en una palabra, pero como hace referencia a una persona deter-

DISPONIBLE
TENISTA
JUGADOR DE CRICKET
LECTOR
FUTBOLISTA
ARTISTA
JUGADOR DE DARDOS
EXPERTO EN VINOS
PROPIETARIO DE COCHE
TRANSPORTE PÚBLICO
CICLISTA

Tony Lodge

¿Alguien juega al tenis?
Hemos utilizado el cubo de Rubik como una analogía de una base de datos rudimentaria. Esto es, una que contiene toda la información que podemos necesitar, pero que aún no ha sido manipulada para obtener el orden correcto. En este ejemplo, se busca un tenista (el símbolo de la raqueta), que posea un coche (símbolo correspondiente) y que tenga tiempo libre el día en cuestión (cuadro rojo).

```
DATABASE  HCCSOFT
ENTER DATE AS DD/MM/YY: 14/10/93

#ACCESS

# use AMIGOS
# while AMIGOS, do COCHES
# while AFICIONES, do TENIS:
# end

.CLOSE
```



Con preguntas adecuadas

La mayoría de programas de base de datos comerciales emplean códigos que por sí mismos son casi lenguajes de programación. El código que hemos usado aquí es característico. "ACCESS" indica al programa que deseamos "interrogar" o formular preguntas al archivo que hemos creado previamente. "Use" indica que vamos a utilizar un subconjunto del archivo "AMIGOS", y "while" (mientras) estas condiciones sean verdaderas, extraeremos todos los ítems etiquetados con "COCHE" o "TENIS". El resultado es una lista de amigos, con sus números de teléfono, que tienen coche y juegan al tenis

Ian McKinnell

Registro

REGISTRO			
NOMBRE		APELLIDO	
		N. DE PILA	
DIRECCION PARTICULAR			
		CASA	
		CALLE	
		CIUDAD	
		PROVINCIA	
		DIST. POSTAL	
DIRECCION PROFESIONAL		OCUPACION	
		CASA	
		CALLE	
		CIUDAD	
		PROVINCIA	
		DIST. POSTAL	
			NOMBRE DEL DIRECTOR
COCHE		MARCA	
	AÑOS		
AFICIONES		EN CASA	
		AL AIRE LIBRE	
PERFIL	EDAD		
		SEXO	
		ESTADO CIVIL	
			CONYUGE

CLAVE: LISTA · NUMERO · PALABRA · INDICADOR

Índice personal

- APELLIDO
- NOMBRE DE PILA
- DIRECCION
- TELEFONO PART.
- OCUPACION
- DIRECCION PROF.
- TELEFONO PROF.
- NOMBRE DIRECTOR
- EDAD
- SEXO
- AFICIONES
- ESTADO CIVIL
- MARCA DEL COCHE
- AÑOS DEL COCHE

Información organizada
Una base de datos jerarquizada conduce al usuario de una información a otra, ofreciéndole cada vez una elección. Se da por supuesto que no se conoce el contenido

minada, y éste es el tema del archivo, sería útil si esta palabra se pudiera relacionar de algún modo con otro registro.

Puesto que cada registro está en un lugar determinado del archivo, se le asigna un número. Así se puede utilizar el número de registro que describe a esa persona, en lugar de su nombre, para establecer una entrada.

A este tipo de dato se le llama registro "índice". Si se utiliza esta técnica para referirse al director de la empresa en la que trabaja una persona determinada, el resultado es una estructura parecida a la denominada "registro".

La diferencia entre una ficha y una base de datos mecanizada es que la primera sólo se puede clasificar de una forma, generalmente la alfabética.

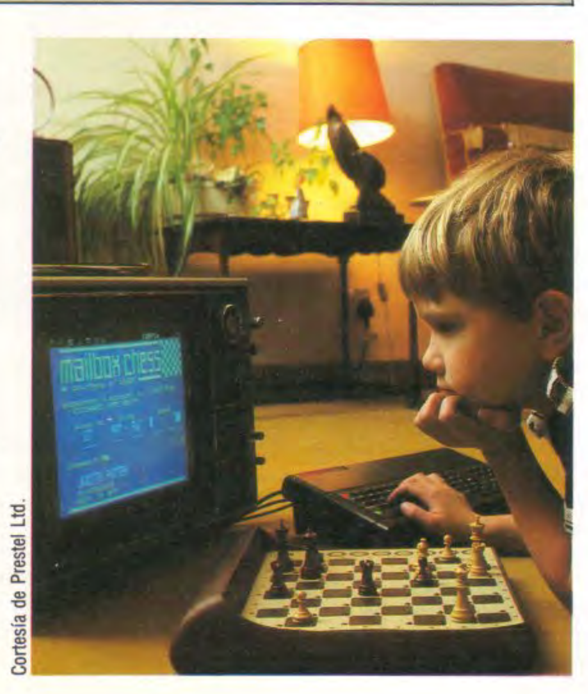
La ficha es adecuada si se quiere saber, por ejemplo, en qué empresa está empleada una persona determinada. Pero ¿qué podemos hacer si lo que se pretende es saber el nombre de todos los empleados de determinada empresa?

Si se utilizan fichas, tendríamos que mirarlas todas y extraer las adecuadas. Hacer esto no sólo requiere tiempo, sino que probablemente será causa de errores.

Con un sistema mecanizado, sin embargo, se le puede pedir a la máquina que busque sucesivamente cada registro y que imprima el nombre de cada persona que trabaja para la empresa en la que estamos interesados.

Por otra parte, se puede hacer que la máquina clasifique el fichero y que los datos referentes a esta empresa sean los más importantes. Se tendría así la misma base de datos, con los mismos datos, pero con una "forma" totalmente diferente. La clasificación situaría todo lo referente a una empresa dada en un grupo, y esta sección daría los nombres de todos los empleados. En fichas, sólo habría un tema prioritario, el apellido. En cambio, en un sistema mecanizado, cualquier apartado puede ser el principal.

De este modo se puede "reformar" la base de datos. Por ejemplo, el tema principal puede ser los coches que poseen estas personas, o el nombre de las ciudades en que viven. Ésta es la gran ventaja que ofrece el sistema de base de datos mecanizado.

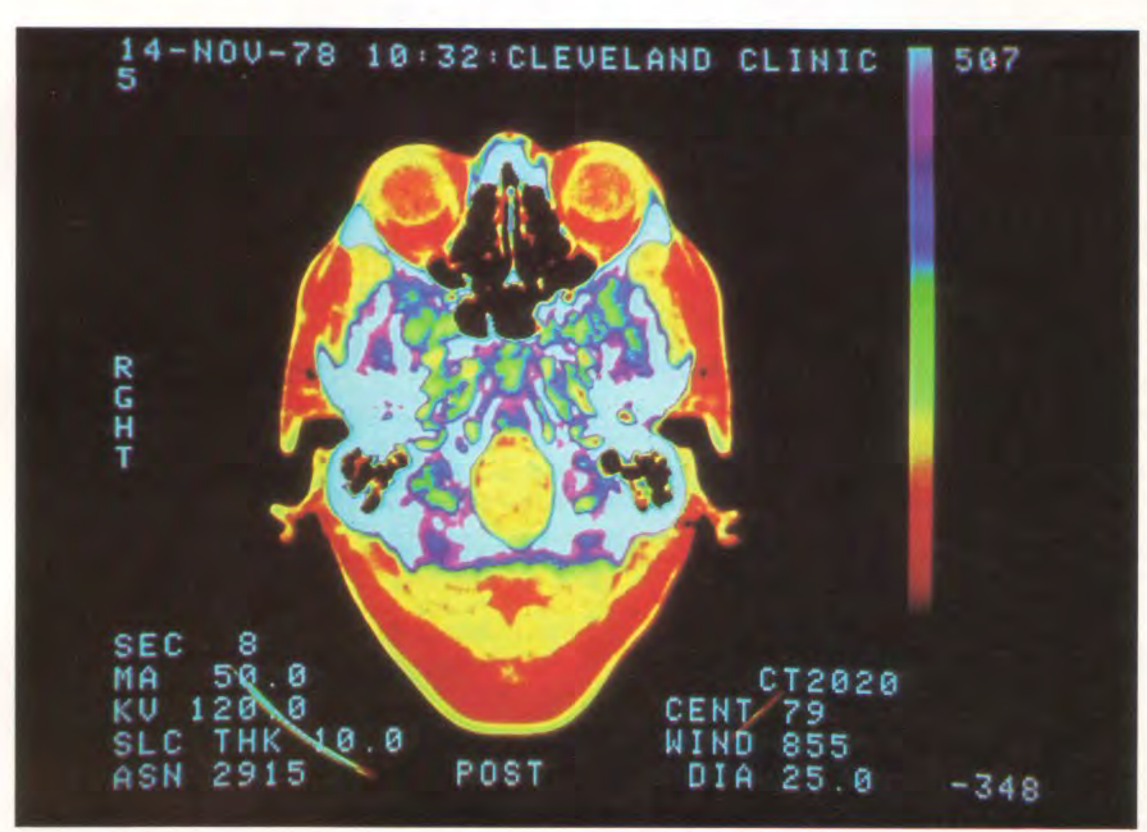


Cortesía de Prestel Ltd.

La enciclopedia de nuestros días
Los servicios del tipo Prestel son un intento de hacer accesibles al gran público bases de datos de gran tamaño mediante servicios que los usuarios ya poseen en sus casas. Los sistemas de videodatos usan el televisor doméstico como monitor, y mediante un teclado similar al de un microordenador, pueden conectar con el ordenador central a través de las líneas telefónicas ordinarias. Los distintos servicios que ofrece la base de datos se muestran en la pantalla, y el usuario llega a la "página" deseada de información siguiendo el organigrama. Introduciendo el código de la tarjeta de crédito, incluso se pueden hacer compras desde casa

Micros en medicina

Una ventaja inapreciable: el ordenador ha reducido sensiblemente la cantidad de "trabajo trivial" que absorbe el precioso tiempo de médicos y enfermeras



Una porción de vida
Con los rayos X convencionales se obtienen representaciones planas, de dos dimensiones, en las cuales están superpuestos todos los órganos. Estas figuras requieren la interpretación de un experto. En cambio, mediante una exploración del cuerpo con un haz de rayos X, recogida de datos con una serie de sensores y el subsiguiente empleo de un ordenador para procesar las "señales" se puede lograr una imagen mucho más precisa de una "porción" del cuerpo humano.
Aunque el ordenador es capaz de producir imágenes en color, tal como se ilustra aquí, muchos radiólogos continúan prefiriendo las reproducciones monocromáticas, pues consideran que éstas dan una representación más clara de las densidades relativas de los tejidos del cuerpo

Al igual que en otras muchas profesiones, en las que el personal altamente cualificado —y muy bien remunerado— utiliza gran parte de su esfuerzo en tareas relativamente triviales, la medicina se ha beneficiado considerablemente de aparatos tan versátiles y económicos como el microordenador.

Poco después de la introducción generalizada de los mecanismos basados en el microprocesador, en particular en las unidades de cuidados intensivos, el método de trabajo cambió de forma considerable. Pronto los microordenadores fueron empleados corrientemente como monitores para el pulso, la respiración y la presión sanguínea, dando lecturas instantáneas de numerosos parámetros y permitiendo a las enfermeras ocuparse de otros cometidos más importantes.

Otros adelantos recientes hacen posible que la administración de un hospital dependa en mayor medida del ordenador: conservación de los registros de los pacientes, notificación de horas de consulta y control de los stocks de medicamentos.

En un apartado anterior (páginas 72 y 73) se mencionó una aplicación importante de los ordenadores en el campo de la medicina, pero merece la pena revisar con mayor detalle cómo se utilizan los sistemas Expert en la práctica.

Mickie, un sistema sencillo pero eficaz que proporciona a los médicos de medicina general información sobre el estado físico actual de los pacientes, es en realidad un Expert System "inexperto", del que no se espera que dé un diagnóstico exhaustivo, sino que suministre sólo una información general.

Las preguntas que el sistema hace a los pacientes pueden ser contestadas siempre con un "sí" , "no", "no sé", "no comprendo". Si la contestación es "no entiendo", el sistema intenta ayudar al paciente a encontrar una respuesta.

El paciente no tiene que responder a través de un teclado completo, sino mediante un mando con sólo cuatro teclas, adecuadamente etiquetadas. El funcionamiento del sistema es bastante lento, pero esto es un límite artificial, ya que se adecua a la velocidad de lectura media. El próximo paso en el desarrollo de un sistema Expert probablemente será relacionar los nuevos datos con el historial médico. Si el paciente se queja de los mismos dolores o expone idénticos síntomas que en una visita anterior, el médico sólo tendrá que hacer una pregunta para efectuar su diagnóstico: "¿Se siente igual que...?"

Algunas de las ventajas que ofrecen los registros médicos mecanizados, tal vez a primera vista no apa-



Un ojo que nunca duerme
A la vez que ofrece al personal hospitalario una indicación del estado actual del paciente, el sistema de monitores puede también almacenar información para un análisis posterior. En la ilustración, un médico obtiene información fidedigna sobre el estado físico de un paciente durante la noche. Esto le ayudará a establecer un diagnóstico más preciso

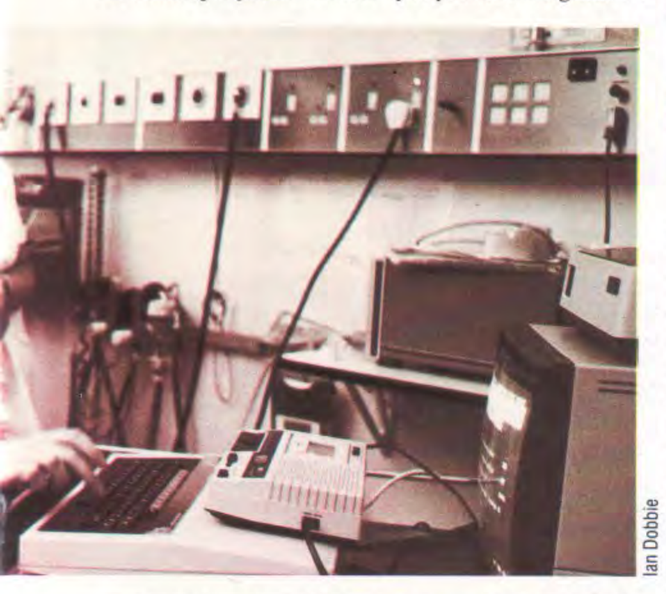
rezcan como tales (por ejemplo, seguridad de información). Es mucho más difícil leer un archivo contenido en un disco magnético que otro escrito en papel, ¡incluso dando por supuesto que se sabe manejar el ordenador!

Pero, con mucha diferencia, los avances más significativos se han logrado en el área del diagnóstico. Hasta hace poco tiempo, los únicos métodos disponibles para un examen interno eran, cuanto menos, peligrosos, e incluso a veces notoriamente destructivos: rayos X, con el peligro derivado de una exposición excesiva a las radiaciones; endoscopia —inserción de un tubo a través de un orificio—, que puede causar lesiones en los frágiles tejidos internos; y cirugía exploratoria.

El desarrollo de técnicas tomográficas mecanizadas, en las que se utiliza un delgado haz exploratorio de rayos X en vez de exponer una parte del cuerpo a una radiación generalizada, ha representado un avance significativo tanto por su precisión como por su seguridad. El empleo de medios alternativos, tales como ultrasonidos y resonancia magnética nuclear, ofrece la posibilidad de efectuar exámenes internos de una forma totalmente indolora.

También los sistemas médicos de urgencia se benefician de la mecanización. Es improbable, por ejemplo, que el programa de trasplante de órganos estuviera tan avanzado sin los recursos que aportan los métodos de base de datos.

En el tratamiento físico y mental de los disminuidos físicos también se ha producido un avance espectacular. Aunque todavía estén sin desarrollar, los adelantos potenciales en este campo son muy importantes. Por ejemplo, cabe imaginar la alegría de un parapléjico mudo que pueda "hablar" por primera vez gracias a

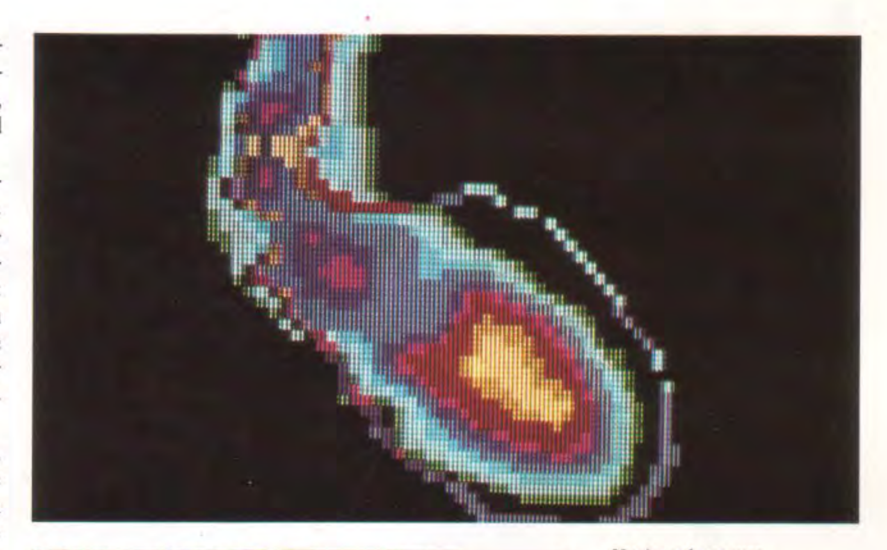


Ian Dobbie

un dispositivo de entrada de ojo-seguidor y salida de sintetizador de voz.

Los dispositivos de ojo-seguidor miden el movimiento que efectúa el globo del ojo al leer a lo largo de una línea. A partir de un punto de referencia, es posible deducir la posición en un momento dado, y por tanto saber el carácter que está leyendo.

Los chips del sintetizador de voz contienen, en memoria ROM, las partículas básicas, llamadas fonemas, con las que se construye el habla. Los fonemas son unidos entre sí, bajo el control del programa, para formar palabras.



Mucho más seguro
Las exploraciones por resonancia magnética nuclear, semejantes a la que se representa aquí, dan unos resultados similares a la tomografía computerizada, pero, al no utilizar rayos X, son considerablemente más seguras. Por el momento, esta técnica se halla en una fase experimental. Cada exploración requiere un tiempo de una hora aproximadamente



Ian Dobbie

Una imagen de salud
La información sobre el estado de un paciente, obtenida mediante sensores colocados en el cuerpo, se interpreta con un microprocesador y se representa en un osciloscopio especialmente modificado, similar al utilizado por los ingenieros electrónicos

Si un minusválido profundo conserva cierto grado de movilidad, ésta puede ser utilizada como base de un sistema de comunicación mecanizado. La forma más fácil de hacerlo es sustituir el teclado estándar por una unidad con un diseño específico. Por ejemplo, ésta podría tener teclas mucho más largas para hacer posible su utilización por personas que padezcan ataxias locomotrices o esclerosis múltiple y que tengan dificultad en la localización de objetos. O bien podrían construirse de forma que no haga falta ejercer presión alguna para manipularlas.

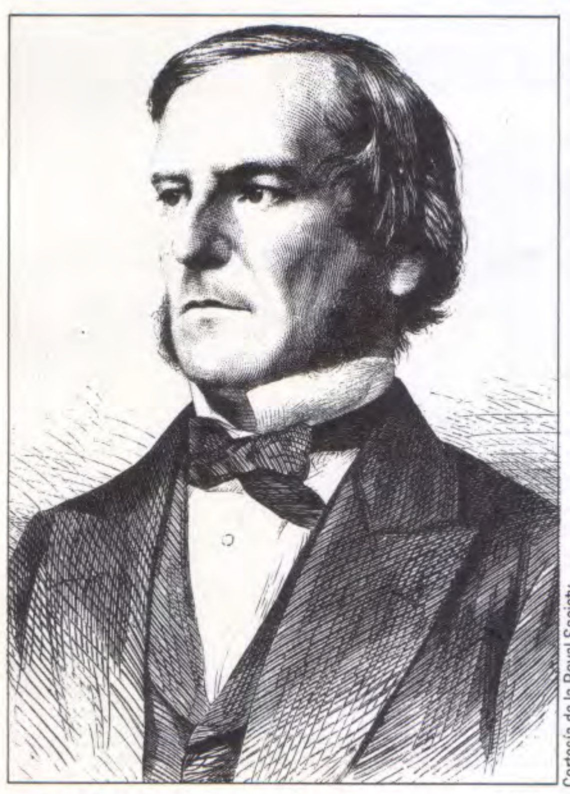
Por otra parte, el ordenador puede ofrecer al usuario una gama predeterminada de servicios; por ejemplo, un menú para la próxima comida. Al apretar una tecla, el usuario dice al ordenador qué quiere comer. En este caso el teclado puede ser un simple interruptor, manejable por ejemplo mediante un simple movimiento de cabeza.

Sin excesiva dificultad, este sistema permitiría al usuario emplear una unidad de tratamiento de textos, aunque lentamente, y así producir una obra escrita, lo que constituiría un avance enorme para aquellas personas que sufren parálisis espasmódica o han sido víctimas de la talidomida.

Así, pues, como en otros muchos campos de la actividad humana, también en medicina el micro se está convirtiendo en una herramienta indispensable.

Las leyes del pensamiento

Un siglo antes de que se inventara el ordenador, George Boole publicó sus ideas sobre la lógica matemática


Cortesía de la Royal Society

George Boole (1815-1864)
George Boole, que nació antes de que los ordenadores electrónicos fueran incluso imaginados, es uno de los fundadores de la lógica matemática usada por los ordenadores actuales. Era hijo de un zapatero remendón y aprendió matemáticas por sí mismo en sus ratos libres. Estaba convencido de que las decisiones que las personas toman cada día se basan en la razón, y de que ésta puede expresarse en términos de lógica matemática. Boole publicó sus ideas en 1847 y, casi inmediatamente, se convirtió en un personaje famoso. Llegó a ser el primer profesor de matemáticas en la entonces recién creada Universidad de Cork

En 1815, el año de la derrota de Napoleón en la batalla de Waterloo, ocurrió otro acontecimiento significativo para la historia de la humanidad. En efecto, en dicho año nacía en Lincoln, Inglaterra, George Boole, hijo de un zapatero remendón que habría de convertirse en uno de los genios que harían posible la invención del ordenador. Aunque murió en 1864, un siglo antes de que empezara la revolución del microordenador, sin sus ideas no hubiera sido posible el desarrollo del ordenador moderno.

Boole sabía que los procesos de razonamiento que las personas efectúan normalmente pueden ser descritos en términos de la lógica formal, de la cual fueron precursores los griegos. Creía que si se intentaba firmemente, se podía llegar a expresar el razonamiento humano en términos matemáticos. Boole se puso a hacer exactamente esto; aprendió matemáticas por sí mismo y empezó sus investigaciones sobre la lógica de las decisiones humanas.

Conjuntos de información

Imagínese que una noche acude a una fiesta. Le apetece bailar y por ello va a la sala y busca una pareja.

La gente de esta sala o bien está bailando o no lo está: no puede hacer ambas cosas a la vez. La pareja a la que se acerca o es un chico o una chica. Obviamente, una persona puede ser hombre o mujer, pero no las dos cosas.

Boole habría enfocado el problema de otra forma. Hubiera visto una pista de baile que incluía "conjuntos" de gente: el conjunto de hombres y el conjunto de mujeres, o H y M. Boole hubiera visto también B y E, o el conjunto de personas que bailan y el de las que esperan para bailar.

Lógicamente, su pareja tendría que satisfacer dos condiciones: ser del sexo opuesto y estar también esperando para bailar. Boole se dio cuenta de la importancia del "y" que relaciona las dos condiciones y le atribuyó un símbolo: una ∪ invertida. Entonces pudo catalogar el conjunto de posibles parejas de baile como $M \cap E$ (o $H \cap E$).

Sin embargo, si esta persona no quisiera bailar, sino sólo charlar con alguien, podría elegir a cualquiera de H o M, porque estos dos conjuntos comprenden a todas las personas de la sala. Otra vez, Boole vio la importancia de la aparentemente inocente "o" de la premisa y le dio el símbolo $\cup$. Así, en su álgebra lógica $H \cup M$ incluye a todos los hombres y mujeres presentes en la sala.

Las puertas lógicas de los ordenadores se designaron como los símbolos de Boole: AND (y) y OR (o). En programación BASIC, descubriremos pronto dos órdenes muy útiles llamadas AND y OR. Pero, volviendo sobre el tema, hay una interpretación muy pintoresca de la lógica booliana, inventada por los matemáticos ingleses John Venn (1834-1923) y Charles Dodgson (1832-1892), este último más conocido como Lewis Carroll, seudónimo con el que publicó numerosas obras para niños.

Veamos un ejemplo práctico. Imaginemos que en la

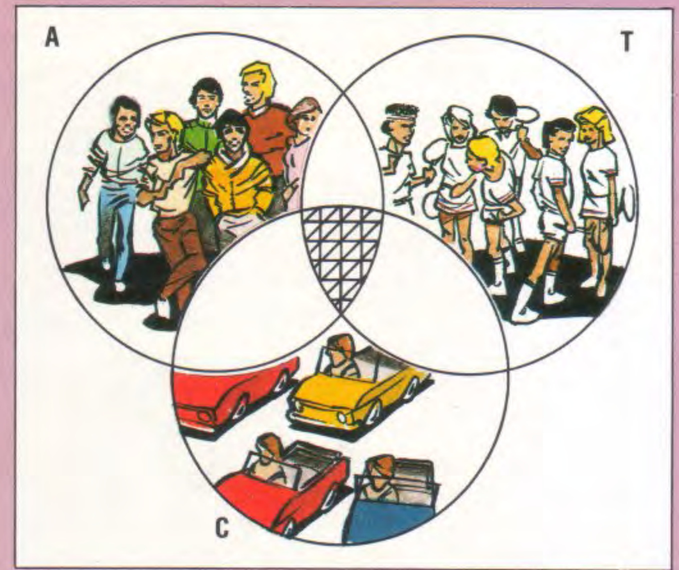
El programa "Amigos"

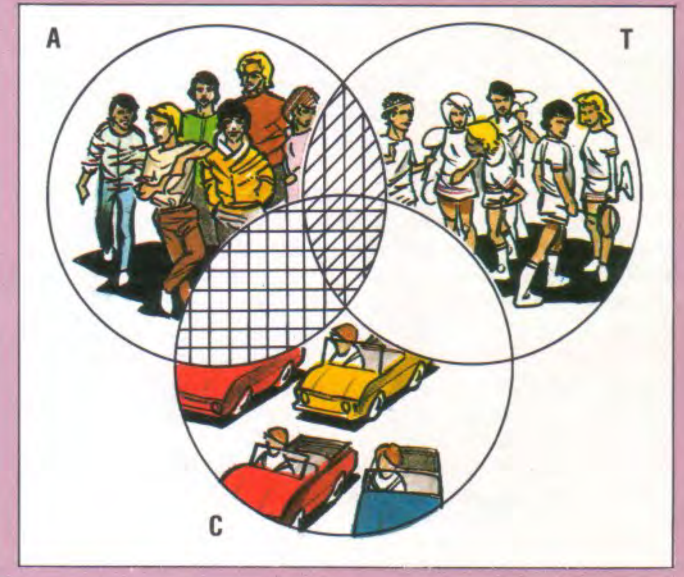
```
10 DIM N$(10), D$(10), F$(10), T$(10), C$(10)
15 REM NOMBRE, NO. TEL., ¿AMIGO?, ¿TENIS?, ¿COCHE?
17 PRINT "INTRODUCIR DETALLES EN LA FORMA":
18 PRINT "NOMBRE, TELEFONO, SI/NO, SI/NO, SI/NO"
20 FOR K=1 TO 10
30 INPUT N$(K), D$(K), F$(K), T$(K), C$(K)
40 NEXT K
45 REM RUTINA BUSCAR
50 FOR J=1 TO 10
60 IF F$(J)="SI" AND T$(J)="SI" AND C$(J)="SI" THEN GOSUB 100
70 NEXT J
80 END
100 PRINT N$(J), D$(J)
110 RETURN
```

Diagramas de Venn









El recuadro representa la colección, o conjunto, de todas las personas listadas en el ordenador. Generalmente, en los diagramas de Venn, éste recibe el nombre de conjunto universal. Los círculos representan los conjuntos individuales. Los amigos se identifican como las personas que figuran en el conjunto A. No todas las personas juegan al tenis. Las que sí lo hacen se incluyen en el conjunto T. Como algunas cumplen ambas condiciones, los dos círculos se superponen (A∩T). El conjunto C identifica a aquellos que poseen coche. Este conjunto intersecciona con los otros dos, y las personas que cumplen las tres condiciones están situadas en ese área (A∩T∩C). El último diagrama representa a los mismos conjuntos (A,T,C), pero las condiciones que se deben cumplir han cambiado. Ahora las premisas son jugar al tenis o ir a dar una vuelta en coche. El conjunto de conocidos que son amigos y juegan al tenis es el representado por diagonales y verticales. El de conocidos que son amigos y poseen coche es el cuadriculado. La intersección de ambas superficies representa a los amigos con los que se podría jugar al tenis o ir a dar una vuelta en coche

Kevin Jones

memoria de su ordenador ha almacenado una lista de las personas que conoce. Con cada nombre se incluye otra información; por ejemplo: número de teléfono, aficiones, etc. Una tarde decide que quiere jugar un partido de tenis en una pista situada en el otro extremo de la ciudad. Para ello necesita un amigo (en contraposición al concepto conocido) que juegue al tenis y que tenga coche. Su ordenador recibe las instrucciones para que escriba el nombre y el número de teléfono de todas las personas que cumplan las tres condiciones: juega al tenis Y tiene coche Y es un amigo.

El programa de la página anterior analiza, primero, la información sobre cada conocido: ¿es amigo?, ¿tiene coche?, ¿juega al tenis? Se supone que usted tiene 10 conocidos, pero puede variar su número según desee (recordando cambiar el 10 entre paréntesis de la sentencia DIM en la línea 10). La lista es examinada por la orden IF...THEN, en la cual se ha insertado una condición múltiple. La mayoría de programas BASIC permiten a la orden IF...THEN trabajar sobre una premisa formada por subcondiciones unidas por AND y OR. Por último, aparecen en la pantalla el nombre y el número de teléfono de todos los conocidos que reúnen las condiciones requeridas: ser amigo, jugar al tenis y tener coche.

En algunos programas se hallan combinaciones muy complejas de funciones lógicas. El álgebra de Boole, que en su tiempo representó poco más que una curiosidad, ha cobrado su verdadera importancia en la era del ordenador.

El Lynx

Este micro de elegante diseño posee una atractiva gama de gráficos y uno de los mejores teclados que existen

El Lynx es un ordenador fabricado por la firma Camputers, de Cambridge (Gran Bretaña). Este micro, aún no disponible en el mercado español, es más grande y pesado que sus dos principales competidores británicos, el Sinclair Spectrum y el Oric, aunque algo más pequeño que los Commodore y Atari, de fabricación norteamericana. Es, asimismo, uno de los ordenadores de aspecto más atrayente, con un funcional acabado en tonalidades grises.

Dispone de una RAM de 48 Kbytes, ampliable hasta 192 Kbytes. Si se considera que un ordenador profesional de gestión posee normalmente al menos 64 K de memoria, resulta evidente la gran versatilidad del Lynx.

Su teclado es similar al de una máquina de escribir tanto en disposición como en tamaño, y ofrece un excelente feedback táctil (se sabe instintivamente si se ha pulsado la tecla correcta).

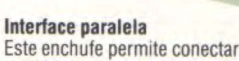
El Lynx puede visualizar ocho colores diferentes y tiene almacenados en su memoria 32 caracteres para gráficos, si bien éstos no están indicados en el teclado. Viene con su propia versión de BASIC, que incluye varias órdenes muy útiles, entre ellas AUTO, que numera automáticamente las líneas del programa.

La unidad central de proceso del Lynx es el microprocesador Z80, que ha sido utilizado en varios microordenadores muy sofisticados.



John Shirreff
El creador de la gama de ordenadores Lynx es John Shirreff, de 35 años de edad. Al poco tiempo de dejar la Universidad de Cambridge, a finales de los años sesenta, y a la vez que actuaba como batería de algunos grupos de rock, diseñó y construyó estructuras portátiles inflables. Continuó trabajando como ingeniero de sonido y como músico mientras se acentuaba su interés por la electrónica. Ambas inquietudes se aunaron al conocer a Dave Vorhaus, quien dirigía un estudio de grabación. En 1981, basándose en el Z80, diseñó un nuevo ordenador de gestión de 64 Kbytes ampliables. La experiencia de este proyecto con el procesador Z80 fue el punto de partida que le llevó a desarrollar la gama de ordenadores Lynx

El teclado del Lynx
Con su carcasa de plástico duro, el Lynx ofrece un aspecto muy pulcro. Posee un teclado profesional con 57 teclas y barra espaciadora. Las teclas alfanuméricas se disponen según el trazado QWERTY estándar. El par de teclas con flechas a ambos extremos de la tercera hilera contando desde arriba, constituyen el mando del cursor y se utilizan sólo con el editor de pantalla. Las teclas ESCAPE y CONTROL están arriba a la izquierda, y las teclas BREAK y DELETE arriba a la derecha



Interface paralela
Este enchufe permite conectar periféricos de conexión en paralelo, como unidades de disco

Enchufe red
Este enchufe DIN conecta el ordenador a la fuente de alimentación eléctrica

Altavoz
Todos los sonidos que genera el Lynx los produce este altavoz



Conexión teclado
Un cable plano conecta el teclado a este tablero de circuitos principal del ordenador, para que éste pueda determinar cuál es la tecla que se ha pulsado

RAM
Estos chips proporcionan la memoria de acceso directo de que dispone el usuario para almacenar programas e información

Microprocesador
La unidad central de proceso del Lynx es un Zilog Z80A



Chris Stevens

Conexión cassette
Este enchufe DIN permite conectar al ordenador una unidad de cassette

Conexión RS232
Este enchufe permite conectar periféricos de comunicaciones en serie, como un modem o un acoplador acústico

Interface RGB
Mediante este enchufe se puede conectar un monitor en color

Modulador
Convierte la señal del chip video de forma que pueda ser aceptada por un aparato de televisión

Reloj
El "latido" electrónico de este oscilador se utiliza para regular y sincronizar todas las operaciones del ordenador

Chip video
Este chip genera la señal de video que puede alimentarse directamente a la interface RGB, para la visualización en un monitor en color, y a un modulador cuando se usa un televisor

Unidades ROM en BASIC
El BASIC del Lynx está almacenado con carácter permanente en este par de unidades ROM

Chip input/output
Este chip convierte las entradas al ordenador en su forma interna, y las salidas de ésta en una forma adecuada para el dispositivo receptor

LYNX

DIMENSIONES

350 × 213 × 60 mm

PESO

1 564 g

CPU

Z80A

VELOCIDAD DEL RELOJ

4 MHz

MEMORIA

RAM de 48 Kbytes, ampliable hasta 192 Kbytes
ROM de 16 Kbytes que proporciona BASIC y monitor (sistema operativo)

VISUALIZACIÓN EN VIDEO

Modalidad de textos de 24 líneas de 40 caracteres. Modalidad de alta resolución con 248 × 256 puntos

INTERFACES

Conexión televisión, conexión cassette RS232, video RGB, enchufe de conexión en paralelo

LENGUAJE SUMINISTRADO

BASIC

VIENE CON

Conexión antena televisión, conexión cassette, unidad de alimentación eléctrica, manual, cassette

TECLADO

Profesional, QWERTY, con 57 teclas y barra espaciadora

DOCUMENTACION

A pesar de que el manual incluye una aceptable introducción al lenguaje BASIC, carece de la clase de información que el usuario más experimentado desearía encontrar. Las explicaciones acerca de la gama de gráficos y el sonido del Lynx son apropiadas para los principiantes; no obstante, la introducción al lenguaje máquina y la guía para el uso del monitor son demasiado inconsistentes para quienes se inician en la informática. A menudo se agrupan temas que no tienen relación alguna. Esto, unido a la inexistencia de un índice, dificulta la localización de algunos ítems. Las ilustraciones corresponden a reproducciones directas de diseños del autor. Si bien esto es aceptable para los dibujos, no lo es para las ilustraciones de carácter técnico

Alta resolución

Ahora es posible comprar una pantalla para el ordenador, con una alta calidad de resolución para gráficos y juegos

Debido a que cada vez es mayor el número de personas que empiezan a usar ordenadores, videos y otros equipos que necesitan pantalla, los precios de los monitores especializados tienden a bajar. En los primeros años, un buen monitor en color no bajaba de las 150 000 pesetas, pero en la actualidad es posible encontrarlo por la mitad de dicho precio. Obviamente, los monitores monocromáticos son mucho más baratos.

Dado el desarrollo constante de las posibilidades gráficas de los microordenadores, muchos de los cuales son en color, adquirir un monitor es una idea muy acertada.

Existen dos tipos principales de monitores en color: el conocido como RGB (*Red, Green, Blue:* rojo, verde, azul), y el video compuesto. El monitor RGB se controla directamente con los tres cañones de electrones, que forman los colores según las indicaciones del ordenador. Los pulsos que se utilizan para sincronizar el ordenador con el monitor también son producidos directamente por el ordenador.

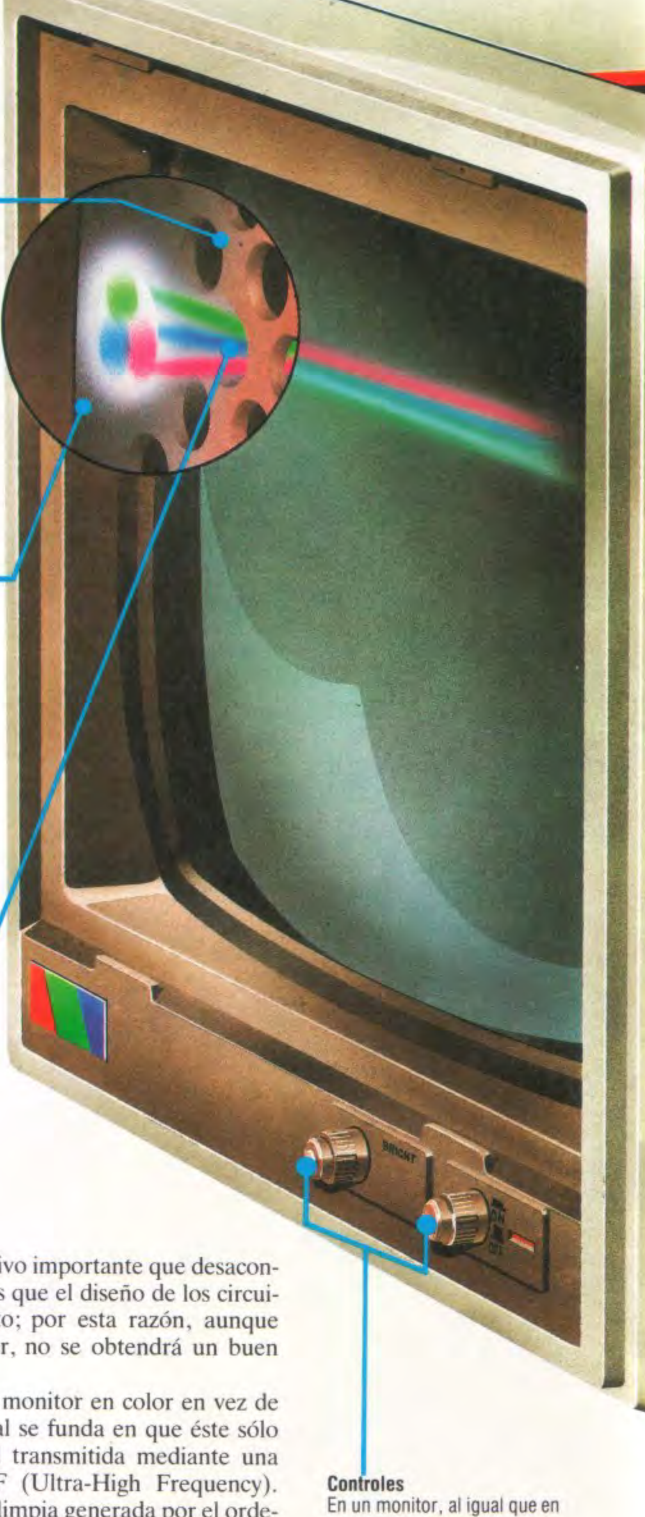
Hay dos tipos de pulsos de sincronización: uno para cada línea de la imagen, y el otro para cada imagen completa. Al final de cada campo, el monitor recibe un pulso, que le dice que se ha llegado al final de la pantalla, y por ello el haz de electrones (y en consecuencia el punto que origina) debe volver a la esquina superior izquierda de ésta.

Al final de cada línea se produce un proceso similar, que indica que esa línea determinada ha sido completada y que el haz de electrones debe regresar al lado izquierdo de la pantalla, para empezar la línea siguiente. En un monitor RGB, cada una de estas señales (roja, verde, azul, sincronismos de línea y de campo) es enviada al monitor mediante cables independientes.

En cambio, un monitor compuesto se asemeja más a un televisor, puesto que todas las señales se combinan en una sola, siendo enviadas al monitor a través de un cable coaxial. Una vez en el monitor, el sincronismo de línea, el de campo y las tres señales del color son de nuevo separados y utilizados para controlar la imagen.

Un monitor es un televisor sin sintonizador. De hecho, se puede transformar un monitor en un televisor añadiéndole un sintonizador, o modificar un televisor normal suprimiendo el mecanismo de selección de canales.

Sin embargo, es totalmente desaconsejable hacer esta adaptación, puesto que en todo componente de un equipo que contenga un tubo de rayos catódicos se generan altos voltajes muy peligrosos. Incluso los técnicos profesionales tratarían este tema con suma precaución.



Rejilla de pantalla
Para asegurar que los cañones de electrones apuntan exactamente al lugar correcto de la pantalla, la superficie del tubo incorpora una rejilla

Fósforo de la pantalla
La imagen en color se compone (tal como se muestra en el diagrama) de tres colores. En el cristal están depositadas varias sustancias que, alcanzadas por el haz de electrones, emiten radiaciones de los tres colores. Al mezclarse, éstos dan las diferentes tonalidades de la imagen, según sea la intensidad del haz en cada punto

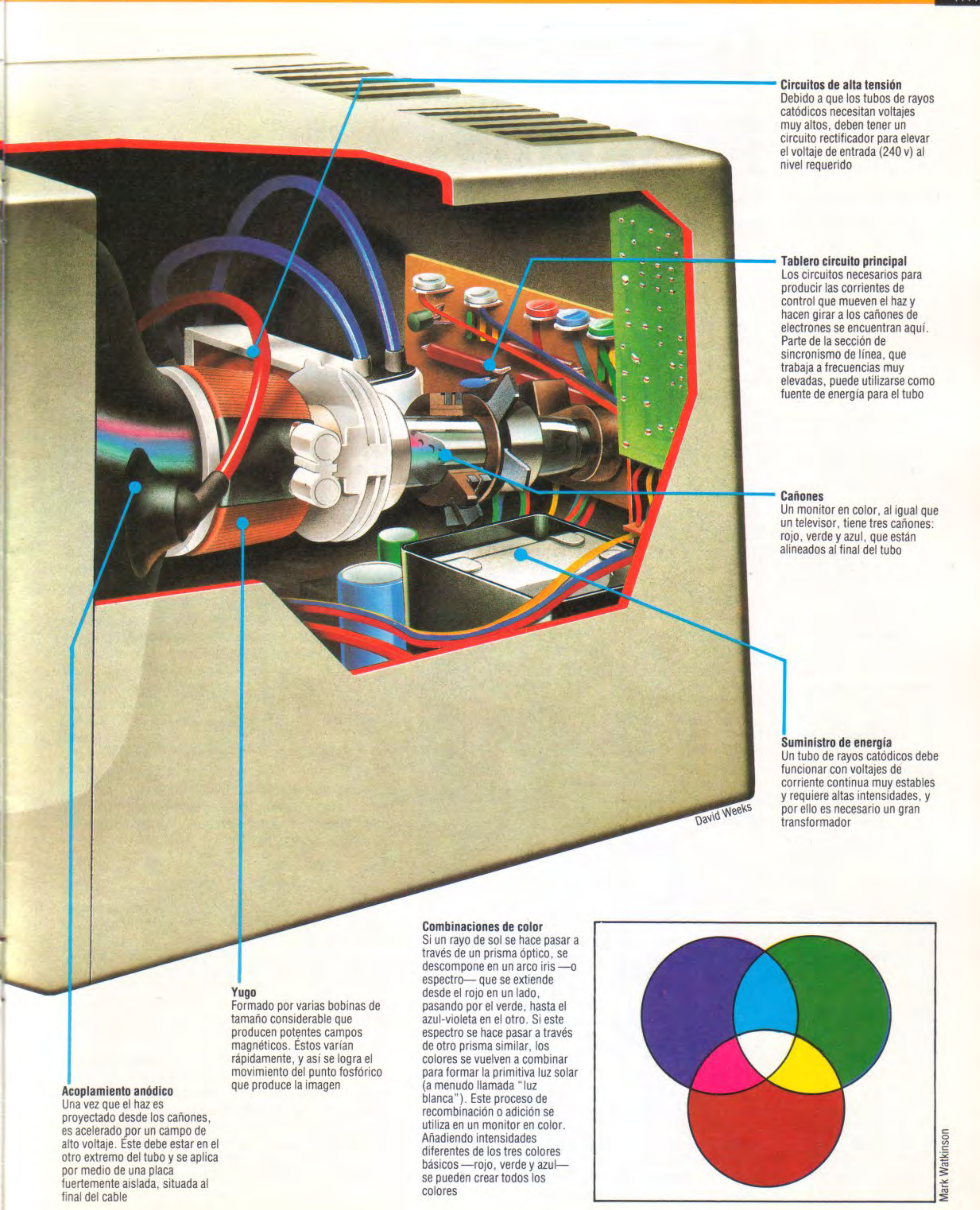
Haces de electrones
En el tubo hay tres haces de electrones, cada uno de los cuales "excita" un elemento de fósforo diferente para producir un punto de color

Controles
En un monitor, al igual que en un televisor, hay varios controles. Normalmente, los de sincronismo vertical y horizontal son accesibles al usuario. La intensidad de color y otras variables, por lo general, no necesitan ser ajustadas y se sitúan en el interior del aparato

Existe además otro motivo importante que desaconseja dicha adaptación, y es que el diseño de los circuitos es ligeramente distinto; por esta razón, aunque se suprima el sintonizador, no se obtendrá un buen monitor.

La razón de utilizar un monitor en color en vez de un televisor como terminal se funda en que éste sólo funcionará con una señal transmitida mediante una onda portadora de UHF (Ultra-High Frequency). Esto significa que la señal limpia generada por el ordenador tiene que ser codificada, enviada a través del cable y decodificada otra vez. Mediante este proceso se recibe una señal "sucia" y, por tanto, se obtiene una imagen borrosa.

Por otra parte, un monitor no necesita esta modulación y demodulación de la señal y la imagen producida será más limpia y precisa. Esto constituirá un descanso para la vista y los programas tendrán un aspecto mucho más profesional.



Circuitos de alta tensión
Debido a que los tubos de rayos catódicos necesitan voltajes muy altos, deben tener un circuito rectificador para elevar el voltaje de entrada (240 v) al nivel requerido

Tablero circuito principal
Los circuitos necesarios para producir las corrientes de control que mueven el haz y hacen girar a los cañones de electrones se encuentran aquí. Parte de la sección de sincronismo de línea, que trabaja a frecuencias muy elevadas, puede utilizarse como fuente de energía para el tubo

Cañones
Un monitor en color, al igual que un televisor, tiene tres cañones: rojo, verde y azul, que están alineados al final del tubo

Suministro de energía
Un tubo de rayos catódicos debe funcionar con voltajes de corriente continua muy estables y requiere altas intensidades, y por ello es necesario un gran transformador

Acoplamiento anódico
Una vez que el haz es proyectado desde los cañones, es acelerado por un campo de alto voltaje. Éste debe estar en el otro extremo del tubo y se aplica por medio de una placa fuertemente aislada, situada al final del cable

Yugo
Formado por varias bobinas de tamaño considerable que producen potentes campos magnéticos. Éstos varían rápidamente, y así se logra el movimiento del punto fosfórico que produce la imagen

Combinaciones de color
Si un rayo de sol se hace pasar a través de un prisma óptico, se descompone en un arco iris —o espectro— que se extiende desde el rojo en un lado, pasando por el verde, hasta el azul-violeta en el otro. Si este espectro se hace pasar a través de otro prisma similar, los colores se vuelven a combinar para formar la primitiva luz solar (a menudo llamada "luz blanca"). Este proceso de recombinación o adición se utiliza en un monitor en color. Añadiendo intensidades diferentes de los tres colores básicos —rojo, verde y azul— se pueden crear todos los colores

Organice su programa

Ordenemos un programa usando funciones preestablecidas para reorganizar la información

Nuestra habitual sección de "Programación Basic" ilustra sobre cómo un programa relativamente complejo puede dividirse en subprogramas sencillos o subrutinas que es posible escribir y verificar por separado.

Además de la ventaja de poder probarlas por separado, el empleo de subrutinas permite desarrollar el programa según una progresión lógica. Existen muchas formas de escribir un programa en BASIC. Una de las más comunes es la denominada "prueba y error", que consiste en empezar a introducir líneas en BASIC en el ordenador sin haber pensado cuidadosamente cómo funcionará el programa. Este procedimiento conduce a programas mal estructurados que no funcionarán la primera vez. Si la estructura del programa no es clara, no resulta fácil encontrar los errores o *bugs*.

Una forma mucho más correcta de abordar el problema sería empezar a trabajar a partir de unas notas previas y elaborar primero la estructura del programa, perfilándolo cada vez con mayor precisión, hasta poder escribir un programa correcto y que funcione. Realizar un diagrama de flujo también ayudará (véase página 104). Veamos cómo se hace.

PROBLEMA: Escribir un programa que introduzca un número de nombres de personas, el nombre de pila seguido por el apellido. Invertir después el orden, de forma que aparezca el apellido en primer lugar, seguido por una coma y un espacio y por el nombre. El programa debe clasificar los nombres por orden alfabético e imprimirlos.

Por ejemplo, si se introducen los nombres BERNARDO TORRES y FRANCISCO ALVAREZ (en este orden), el programa imprimirá:

ALVAREZ, FRANCISCO
TORRES, BERNARDO

Antes de intentar escribir un programa que realice esto, es necesario escribir el input y output deseado en términos muy generales:

Paso 1
Introducir nombres sin orden, primero el nombre de pila
Extraer nombres en orden alfabético, primero apellidos

Esto clarifica lo que queremos que haga el ordenador. Es un primer paso esencial para ordenar adecuadamente el programa. El paso siguiente consiste en afinar las etapas del primero y asegurarse de que el programa funciona. A este nivel, no es necesario detallar en exceso. Simplemente, escribir con algo más de precisión las partes de que consta:

Paso 2
Detallar número de nombres a introducir
Introducir nombres
Invertir nombres
Clasificar nombres
Imprimir nombres

Revisemos la lista anterior y comprobemos si puede funcionar. ¿Hay algo que está mal? ¿Hay algún defecto en la lógica? Si todo está correcto, se puede pasar al siguiente nivel de precisión.

Los apartados del paso 2 son lo suficientemente reducidos y sencillos como para poder escribirlos por separado en subprogramas pequeños. En BASIC, los subprogramas reciben el nombre de *subrutinas*. Asignemos nombres a las subrutinas para identificarlas con mayor facilidad. Subrutina 1, detallar número de nombres a introducir: puede denominarse NUMERO. Subrutina 2, introducir nombres: INTRODUCIR. Subrutina 3, invertir nombres: INVERTIR. Subrutina 4, clasificar nombres: CLASIFICAR. Por último, subrutina 5, imprimir nombres: IMPRIMIR NOMBRES.

Paso 3.1 NUMERO
El operador debe introducir el número requerido
Obtener el número N
Usar N para establecer la longitud del vector del string

Paso 3.2 INTRODUCIR
Si el número de nombres es menor que N, hacer que el operador introduzca otro nombre
Añadir nombre al vector

Paso 3.3 INVERTIR
Hallar longitud del string (nombre)
Encontrar "espacio" en el vector
Colocar caracteres en serie hasta "espacio" en una variable del vector provisional
Colocar caracteres en serie desde "espacio" hasta el final en otra variable provisional
Añadir coma y espacio al final de variable
Asignar segunda seguida por primera variables provisionales al vector original

Paso 3.4 CLASIFICAR
Comparar primer ítem del vector con siguiente
Si primer ítem es mayor que el siguiente (posterior en el alfabeto), intercambiar
Comparar segundo ítem con tercero
Intercambiar, si es necesario
Repetir hasta comparar todos los pares
Retroceder y volver a empezar vector y repetir comparación de pares hasta que no se produzcan intercambios

NOTA: Esta rutina de clasificación es exactamente la misma que la empleada en la parte anterior del curso de programación. La sección "intercambio" *(swap)* será tratada como una subrutina extraída de la subrutina CLASIFICAR.

Paso 3.5 IMPRIMIR NOMBRES
Imprimir cada ítem del vector hasta que hayan sido impresos todos

Cada uno de los pasos necesarios para construir este programa ha sido ya desarrollado con el detalle necesario. La rutina de CLASIFICACION sólo ha sido mencionada superficialmente, puesto que ya se habló de ella en el capítulo anterior de "Programación Basic". Por último, INTERCAMBIAR *(swap)*, que se extrae de la subrutina anterior, ha sido omitida por completo. Veamos ahora qué sencillo resulta convertir programas tratados en lenguaje corriente a un programa en BASIC.

Paso 4
1. NUMERO

Las tres líneas del paso 3.1 traducen directamente en términos BASIC. El operador recibe instrucciones mediante la sentencia PRINT, el número se halla con INPUT y la relación de nombres se dimensiona utilizando la sentencia DIM:

```
PRINT "¿CUANTOS NOMBRES DESEA INTRODUCIR?"
INPUT N
DIM A$ (N)
RETURN
```

Ahora la variable N contiene el número máximo de nombres que se deben introducir. DIM dimensiona un vector. Las variables contienen series de caracteres alfanuméricos, en vez de números. Un nombre de variable siempre termina con un signo "dólar". A$ sola únicamente puede contener un string. DIM A$ (N) crea una relación que puede contener N strings. Las variables subíndices ya han sido estudiadas con anterioridad en este curso.

La sentencia RETURN devuelve el control al programa principal, en la línea siguiente a la que se ha introducido la subrutina. Los valores asignados a las variables serán retenidos por el programa principal y pueden emplearse en cualquier punto de éste, incluso en otras subrutinas.

2. INTRODUCIR

Mientras el número de nombres introducido sea menor que N, hay que recordar al operador que introduzca un nombre, el cual debe ser añadido al vector. Esto hace necesario crear un bucle FOR-NEXT; sabemos que el primer nombre del vector será su primer elemento, y que el último será el enésimo, por consiguiente:

```
FOR X = 1 TO N
PRINT "INTRODUCIR NOMBRE"
INPUT A$ (X)
NEXT X
RETURN
```

Esto debería ser suficiente para introducir todos los nombres, pero el lector atento habrá intuido qué sucede cuando se invierte el orden del nombre y apellido en la subrutina INVERTIR. Cada elemento (nombre) del vector debe ser extraído otra vez, invertido y vuelto a colocar en el vector. En vez de complicar y alargar el programa al hacer esto, sería más sencillo extraer la subrutina INVERTIR del interior de la subrutina INTRODUCIR después de que se haya impreso cada nombre. El nombre puede invertirse así antes de ser asignado al vector. Para realizar esto, sólo se tiene que añadir una línea:

```
FOR X = 1 TO N
PRINT "INTRODUCIR NOMBRE"
INPUT A$(X)
GOSUB [INVERTIR]
NEXT X
RETURN
```

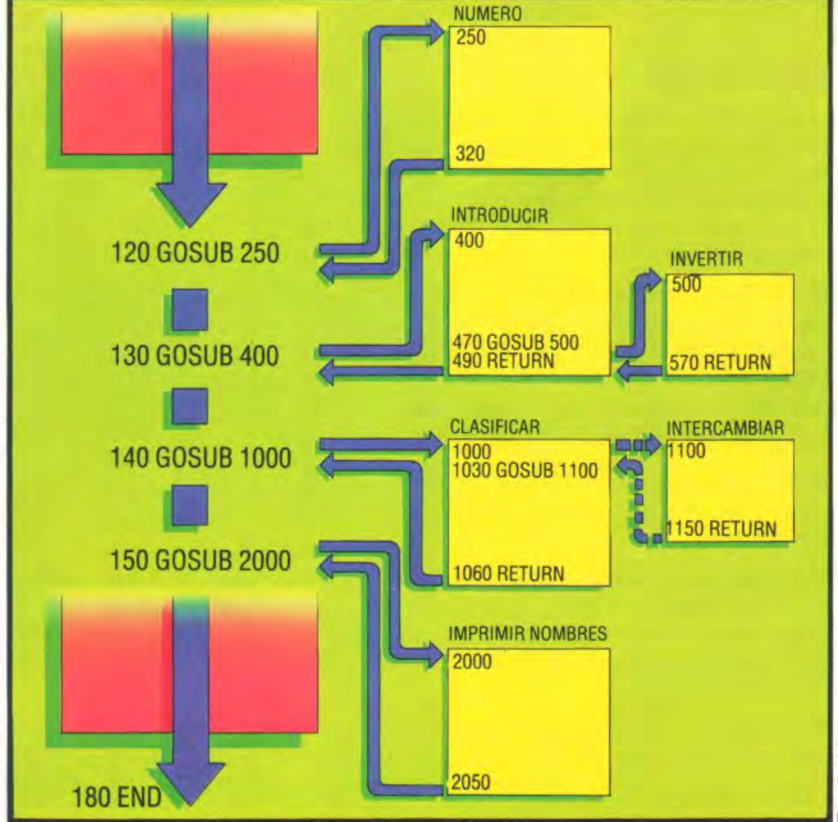
Todos los nombres del vector aparecerán ahora en orden inverso (primero el apellido, seguido del nombre de pila) y, en consecuencia, estarán listos para ser clasificados.

3. INVERTIR

Para invertir el orden de los nombres, necesitamos saber dónde va el "espacio" que separa el nombre de pila del apellido. Cuando sepamos dónde está este espacio, podremos utilizar varias funciones para extraer secciones del vector y asignarlas a otros vectores. Las funciones en lenguaje BASIC consisten en órdenes que realizan una operación predefinida sobre el valor que sigue al nombre de la función. Esta parte está siempre entre paréntesis. Muchas funciones están ya incorporadas, pero también cabe la posibilidad de definirlas uno mismo. Una típica función "preestablecida" es la SQR ( ). Esta función halla la raíz cuadrada del valor entre paréntesis. Así, LET A = SQR(9): PRINT A imprimirá un 3.

INVERTIR utiliza las funciones LEN (para hallar la longitud de la serie), INSTR (para hallar la posición del espacio), LEFT$ (para quitar un número determinado

Programas dentro de un programa
Esta vez el programa principal es muy corto. El auténtico trabajo es realizado por los subprogramas (llamados, en BASIC, subrutinas). Cada uno de los pasos necesarios para que funcione el programa es separado y escrito como un "miniprograma". Luego son unidos por el programa principal.
Al pasar el programa, cada vez que se encuentra una sentencia GOSUB, aquél se desvía hacia el número de línea de la subrutina especificada y se efectúa esa sección del programa. El final de la subrutina se indica con la sentencia RETURN.
Al llegar aquí, el programa vuelve al punto inmediatamente posterior al GOSUB que ha introducido la subrutina.
Las subrutinas, a su vez, pueden "anidar" otras y desviar el programa hacia ellas. Así, INTRODUCIR conduce a la denominada INVERTIR, y CLASIFICAR, a veces, hace aparecer otra subrutina llamada INTERCAMBIAR.
Al dividir un problema en subrutinas independientes, relacionadas por un programa principal sencillo, se logra que el desarrollo y verificación de éste sea mucho más fácil



Tony Lodge

de caracteres de la izquierda del string) y RIGHT$ (para quitar un número determinado de caracteres de la derecha del string). No entraremos en detalles, por el momento, de cómo actúan exactamente estas funciones. En el próximo apartado del curso veremos con más detalle las funciones en BASIC.

4. CLASIFICAR

CLASIFICAR y la subrutina INTERCAMBIAR extraída de ella son muy parecidas a las rutinas utilizadas más arriba.

5. IMPRIMIR NOMBRES

Es muy sencilla:

```
FOR Q=1 TO N
PRINT A$(Q)
NEXT Q
RETURN
```

Ahora lo único que falta es escribir el programa principal. Es tan simple como:

```
REM PROGRAMA PRINCIPAL
GOSUB [NUMERO]
GOSUB [INTRODUCIR]
GOSUB [CLASIFICAR]
GOSUB [IMPRIMIR]
END
```

Los "nombres" de las subrutinas se han puesto entre corchetes. Algunos lenguajes BASIC pueden llamar a las subrutinas por el nombre, pero la mayoría tiene que usar números de línea. Cuando el programa está realmente escrito, los números de línea correspondientes se insertan en lugar de los nombres de las subrutinas. También se añaden los REM adecuados y los mensajes PRINT.

Ejercicios

Ahora que ya se han explicado casi todas las características más importantes del lenguaje BASIC, es el momento de comprobar su progreso realizando estos ejercicios. La dificultad para resolverlos es diferente en cada caso: fluctúa desde lo muy fácil hasta lo moderadamente difícil.

■ **Variables.** Algunos de los nombres de variables que se dan a continuación pueden almacenar valores numéricos y otros no son nombres válidos para emplear como variables. Marcar con un círculo las variables numéricas válidas y tachar las que no lo sean.

A B6 2Z D$ 15 X$ A12 D9 Q81 Q5 6F H$

■ **Aritmética 1.** Escribir un programa corto para asignar el valor 6 a la variable B y PRINT (imprimir) el valor de B.

■ **Aritmética 2.** Escribir un programa corto para asignar el valor 5 a la variable A, 7 a la variable B y 9 a la C. Sumar los valores de estas tres variables y asignar el valor obtenido a la variable D. PRINT el valor de la variable D.

■ **Aritmética 3.** Mire estas líneas de BASIC y calcule cuál ha de ser el valor de C.

```
LET C = 5 + 4 * 3
PRINT C
```

```
10 REM PROGRAMA CLASIFICA NOMBRES
20 REM POR ORDEN ALFABETICO
30 PRINT "PRIMERO DECIDIR CUANTOS"
40 PRINT "NOMBRES DESEA INTRODUCIR"
50 PRINT "LUEGO INTRODUCIR LOS NOMBRES"
60 PRINT "NOMBRE (ESPACIO) APELLIDO"
70 PRINT "ORDEN."
80 REM
90 REM ESTE ES EL PROGRAMA PRINCIPAL
100 PRINT
110 PRINT
120 GOSUB 250
130 GOSUB 400
140 GOSUB 1000
150 GOSUB 2000
160 REM
170 REM FIN DEL PROGRAMA PRINCIPAL
180 END
250 REM SUBRUTINA PARA HALLAR NO. DE
260 REM NOMBRES A INTRODUCIR
270 PRINT "¿CUANTOS NOMBRES"
280 PRINT "DESEA INTRODUCIR?"
290 PRINT
300 INPUT N
310 DIM A$(N)
320 RETURN
400 REM SUBRUTINA PARA INTRODUCIR NOMBRES
410 PRINT "INTRODUCIR NOMBRE EN ESTA FORMA:"
420 PRINT "NOMBRE(ESPACIO)APELLIDO(CR)"
430 PRINT "P. EJ. ANA TORRES"
440 FOR X = 1 TO N
450 PRINT "INTRODUCIR NOMBRE"
460 INPUT A$(X)
470 GOSUB 500
480 NEXT X
490 RETURN
500 REM SUBRUTINA PARA INVERTIR ORDEN
    DE NOMBRES
510 LET L = LEN(A$(X))
520 LET S = INSTR(A$(X)," ")
530 LET C$ = LEFT$(A$(X),S - 1)
540 LET F$ = RIGHT$(A$(X),L - S)
550 LET F$ = F$ + ","
560 LET A$(X) = F$ + C$
570 RETURN
1000 REM RUTINA CLASIFICAR
1010 LET S = 0
1020 FOR P = 1 TO N - 1
1030 IF A$(P)> A$(P + 1) THEN GOSUB 1100
1040 NEXT P
1050 IF S = 1 THEN GOTO 1000
1060 RETURN
1100 REM SUBRUTINA INTERCAMBIAR
1110 LET T$ = A$(P)
1120 LET A$(P) = A$(P + 1)
1130 LET A$(P + 1) = T$
1140 LET S = 1
1150 RETURN
2000 REM IMPRIMIR SUBRUTINA
2010 PRINT
2020 FOR Q = 1 TO N
2030 PRINT A$(Q)
2040 NEXT Q
2050 RETURN
```

■ **Aritmética 4.** ¿Qué resultado se imprimirá en este programa?

```
LET A = 3
LET B = 2
LET C = 9
LET D = 4
LET E = (A + B) * (C − D)
PRINT E

LET E = 5
LET E = E * E
PRINT E
```

■ **Comparaciones 1.** ¿Cuál será el valor de X correcto para que se imprima el mensaje PRINT?

```
 70 LET A = 5
 80 LET B = X
 90 LET R = B − A
100 IF R = 0 THEN GOTO 120
110 GOTO 10
120 PRINT "¡FELICITACIONES! HA GANADO"
999 END
```

■ **Comparaciones 2.** ¿Cuál es el valor menor de X que hace saltar el programa a la línea 300?

```
250 IF X > 6 * 100 THEN GOTO 300
```

■ **Comparaciones 3.** ¿Cuál es el menor valor de Z que hace saltar el programa al mensaje de "felicitaciones"?

```
340 IF Z < 10000 THEN GOTO 500
350 IF Z > = 10000 THEN GOTO 520
:
:
500 PRINT "SU PUNTUACION ES DEMASIADO BAJA.
    INTENTELO DE NUEVO"
510 GOTO 600
520 PRINT "FELICITACIONES. AHORA ES UN
    MAESTRO"
530 GOTO 700
```

■ **Print 1.** Suponga que el valor de T es 50. Escriba una sentencia PRINT que diga: EL VALOR DE T ES 50. Se aconseja colocar el "mensaje" entre comillas, usar un punto y coma y el nombre de la variable.

■ **Print 2.** Mire el siguiente programa y complete la sentencia PRINT de forma que el programa imprima un mensaje semejante a éste:

```
PERDON, PERO SU PUNTUACION DE 175 ES
DEMASIADO BAJA
```

Complete la línea de forma que el valor real de la puntuación pueda variar cada vez.

```
620 REM LA VARIABLE S ES LA PUNTUACION
    HASTA AHORA
620 IF S < = 500 THEN GOTO 640
630 GOTO 700
640 PRINT "LO SIENTO"
```

■ **Print 3.** ¿Qué mensaje se imprimirá una vez se haya pasado el programa?

```
200 LET A$ = "¿MI COMPUTER?"
210 LET B$ = "¿LE GUSTA?"
220 PRINT B$
230 PRINT A$
```

■ **Input 1.** INPUT es una forma de asignar un valor a una variable. Si se pasa el siguiente programa, ¿qué tecla deberá digitarse para que el programa imprima 12 como respuesta?

```
60 INPUT N
70 LET N = N * 2
80 PRINT N
```

■ **Input 2.** ¿Qué se imprimirá aquí?

```
100 PRINT "POR FAVOR DIGITE SU NOMBRE"
110 INPUT N$
120 PRINT "HOLA"; N$; "SOY SU ORDENADOR"
```

Complementos al BASIC

Este programa no funcionará en el Atari 400/800, puesto que su tratamiento de vectores es muy diferente del de las otras máquinas.

DIM

Existe en el Spectrum, pero su uso no es estándar; por ello es necesario suprimir la línea 310 y reemplazarla por:

```
310 DIM A$ (N,30)
```



En la línea 1050, la orden GOTO 1000 viene inmediatamente después de la palabra THEN. En este caso, la mayoría de ordenadores permiten omitir la palabra GOTO; por tanto, la línea 1050 podría escribirse:

```
1050 IF S=1 THEN 1000
```



Spectrum, VIC 20, C64 y Oric-1 no disponen de esta función.
En la serie Commodore y Oric-1, se debe suprimir la línea 520 y reemplazarla por estas cinco:

```
515 FOR P=1 TO L
520 CH$=MID$ (A$(X),P,1)
522 LET S=0
523 IF CH$=" " THEN LET S=P: LET P=L
525 NEXT P
```





El Spectrum no dispone de ninguno de estos mandos, pero pueden crearse versiones propias de ellos con DEF FN; por tanto, suprimir la línea 320 y reemplazarla por las cuatro siguientes:

```
320 DEF FN M$ (X$,N)=X$(N)
330 DEF FN L$ (X$,N)=X$ (TO N)
340 DEF FN R$ (X$,N)=X$ (N TO)
350 RETURN
```

Luego suprimir las líneas 510 a 560 y reemplazarlas por las siguientes:

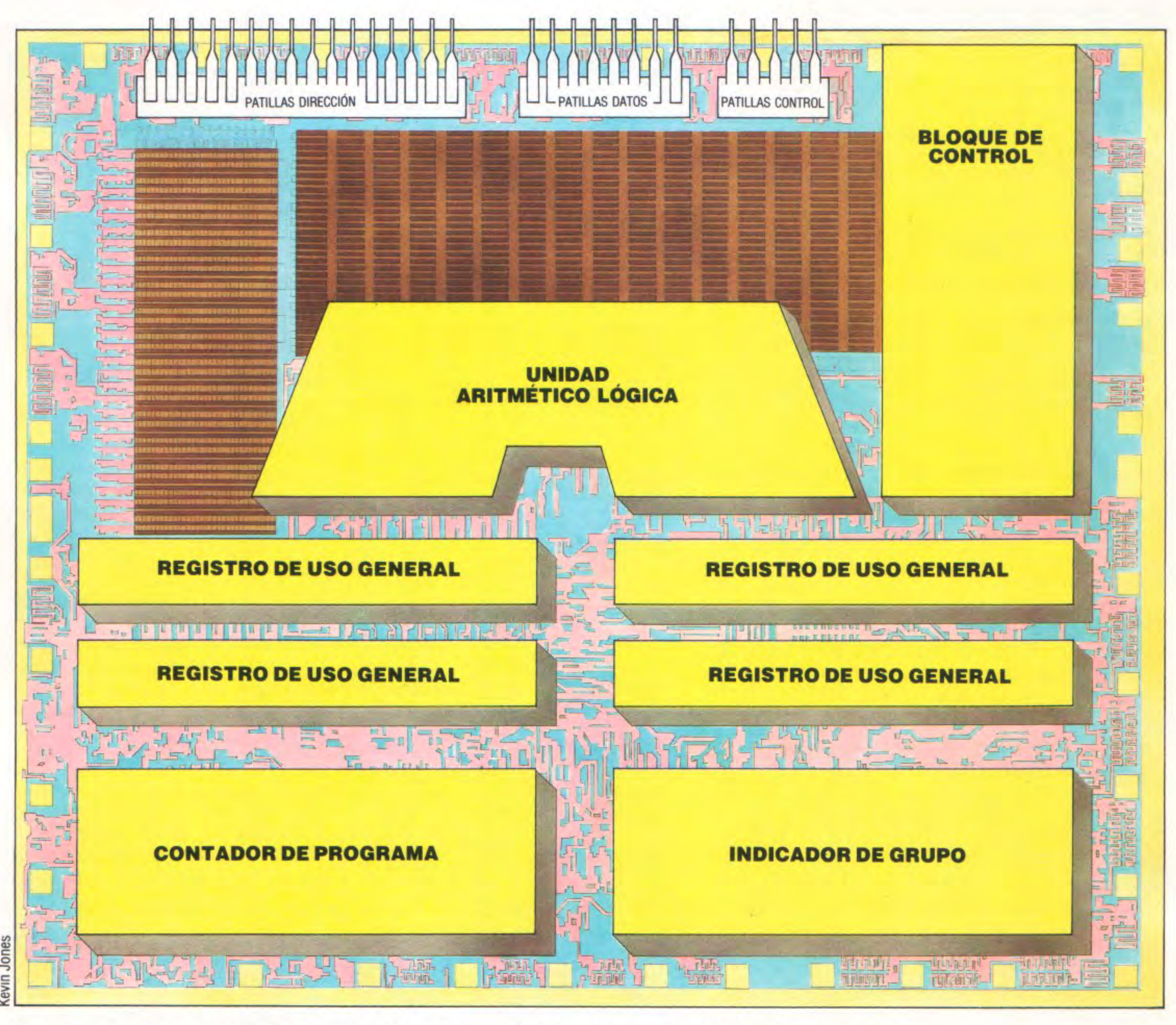
```
510 LET D$=A$ (X)
520 LET L=LEN (D$)
530 LET S=0
540 FOR P=1 TO L
550 IF FN M$ (D$,P)=" " THEN LET S=P
560 IF S<>0 THEN LET P=L
570 NEXT P
580 LET C$=FN L$ (D$,S−1)
590 LET F$=FN R$ (D$,L−S)
600 LET G$=F$+"," + C$
610 LET A$ (X)=G$
620 RETURN
```

END

No existe en el Spectrum, Oric-1 y Dragon 32; reemplazarlo por STOP.

El centro nervioso del ordenador

Todos los canales de la actividad de un ordenador se originan y confluyen en su "unidad central de proceso"



Kevin Jones

El funcionamiento del ordenador es controlado por la CPU (unidad central de proceso). El contenido semántico de estas palabras se puede interpretar casi literalmente: unidad (forma un conjunto independiente); central (está situada en el corazón del ordenador); de proceso (realiza el trabajo). Un ordenador muy simple (véase la ilustración) puede estar formado tan sólo por unos circuitos de CPU, memoria e input/output(I/O).

El ordenador necesita el I/O para comunicarse con el mundo exterior. En una aplicación muy sencilla, un ordenador incorporado a una lavadora automática necesitaría los circuitos de I/O para conectar el motor y los calentadores. La memoria se necesita para almacenar las instrucciones y datos que debe procesar la CPU. Estos datos procesados por la CPU pueden incluir números y códigos binarios que representan caracteres (letras, dígitos y signos tales como @ y !).

Si unas zonas de la memoria contienen instrucciones para la CPU, y otras datos que deben ser procesados por ésta, ¿cómo diferencia ambos conjuntos? Para contestar a esta pregunta, es necesario conocer el interior de un microordenador.

La CPU de los microordenadores de 8 bits (la mayoría de los ordenadores personales pequeños son de este tipo) normalmente está formada por un solo chip de 40 patillas, 20 en cada uno de sus lados. Cada una de estas patillas (excepto las conectadas a la fuente de 0 y +5 v) transporta señales desde o hacia la CPU y otros componentes, por ejemplo los circuitos I/O o los de memoria.

Por lo general, una CPU de 8 bits tiene 16 patillas de dirección, que se conectan al "bus de direcciones". Cada una de estas patillas lleva una señal de salida, que representa un uno o un cero. Se pueden formar 65 536 combinaciones distintas de unos y ceros. Se emplean para seleccionar puntos específicos de la memoria.

También hay ocho patillas de "datos", que están conectadas al "bus de datos". Éstas transportan datos desde la memoria o I/O al interior de la CPU o viceversa.

Otras patillas transportan señales de "control" que funcionan como entradas o salidas de la CPU. Más adelante veremos cómo se utilizan estas señales de control.

Bajo control
La ilustración muestra una CPU con "registros" de memoria, una unidad aritmético lógica, que consta de cientos de puertas lógicas (para realizar las operaciones de adición, AND y de complementación de números binarios), y un bloque de control. Este último admite las instrucciones codificadas (en binario), las interpreta y hace que las otras partes de la CPU procesen adecuadamente. Por ejemplo, si una orden significa que el contenido del acumulador debe almacenarse en un punto determinado de la memoria, el bloque de control pondrá la dirección en las patillas correspondientes, enviará señales de control para que la memoria almacene los datos, y pondrá el contenido del acumulador en el bus de datos para transmitirlo a la memoria

Las celdas de memoria o registros

En el interior de la CPU hay unas celdas de memoria de uno o dos bytes, llamadas *registros*. Algunas de estas celdas de memoria se reservan para fines especiales, y las otras se usan para el almacenaje temporal de información. Estas últimas se denominan *registros de utilización general*. En la CPU existen otros dos "bloques" funcionales importantes: la ALU y el "bloque de control".

La ALU (Arithmetic and Logic Unit: unidad aritmético lógica) realiza las operaciones lógicas y aritméticas que incluyen (pero no se limitan a) adición, operaciones AND y OR, y desplazamiento de bits hacia la derecha o la izquierda dentro de un byte.

Cumpliendo órdenes
Un ordenador muy sencillo puede constar únicamente de una CPU, memoria y un circuito I/O. La memoria almacenará órdenes especiales que harán que la CPU realice acciones específicas. También almacenará datos que serán procesados por la CPU, según las instrucciones. El circuito I/O es necesario para que la CPU se comunique con el exterior. Si el ordenador controla una lavadora automática, el circuito I/O introducirá las señales de los mandos y permitirá que salgan las órdenes para conectar y desconectar el motor y los calentadores.
Los códigos de órdenes para la CPU estarán en sistema binario. Cada modelo diferente de CPU tiene su propio conjunto de códigos

El bloque de control es un circuito especial diseñado para lograr que la CPU se comporte de acuerdo con las instrucciones recibidas desde la memoria. Podemos ver un ejemplo muy explícito utilizando los códigos de instrucciones para la popular CPU del Z80. Si desde la memoria se recibe la instrucción codificada 11000110, la CPU sumará el contenido del próximo byte de la memoria al de uno de los registros del interior de la CPU. Si se quiere almacenar el resultado de esta adición en un punto determinado de la memoria, la siguiente instrucción que reciba la CPU tendrá que ser 00110010, seguida de dos bytes que especifiquen la situación real en la memoria en la cual se puede almacenar el resultado.

Supongamos que el resultado de la adición sea 37 (en notación decimal), y que los dos bytes que determinan la dirección sean 33126 (también en notación decimal). El código de instrucción hará que el bloque de control coloque las patillas de dirección según el equivalente binario de 33126 (éste sería 1000000101100110). Asimismo, hará que las patillas de control envíen señales a la memoria diciéndole que va a recibir datos que deben ser almacenados (memorizados). También motivará que las patillas de datos adopten la disposición del equivalente binario de 37 (00100101). Esta información pasará a través del bus de datos a la memoria y será almacenada en el punto de la misma determinado por el bus de direcciones. Si más tarde la CPU necesitara procesar estos datos de otra forma (impresos en la pantalla, por ejemplo), se podría enviar a la CPU una orden diferente. El bloque de control interpretaría esta instrucción como: "Dirigirse al punto de memoria 33126, tomar el byte que hay en él y almacenarlo provisionalmente en uno de los registros internos".

El número de registros, o celdas de memoria provisional, en el interior de la CPU depende de ésta. Serán registros del tipo 8 bits (un byte) o del tipo 16 bits (dos bytes). Por lo general, los registros especializados reciben nombres determinados; por ejemplo, "indicador de grupo", "contador de programa" o "acumulador". Los registros generales reciben la denominación de "el registro X", "el registro Y", "el registro C", etcétera.

Uno de los registros de 16 bits, y uno de los más importantes, será el "contador de programa". Esta celda de memoria interna contiene siempre la dirección (en binario) de la siguiente instrucción que debe ejecutarse. Cuando llegue el momento de extraer la siguiente orden para la CPU, el contenido del contador de programa será colocado en el bus de direcciones, y el byte correspondiente será transmitido (vía bus de datos) a la CPU.

El registro más importante de 8 bits es el "acumulador". Éste es el que, por lo general, almacena (provisionalmente) el resultado de las operaciones realizadas por la ALU: bytes tomados de la memoria o de I/O, o del lugar donde están almacenados provisionalmente antes de ser enviados a éstos.

Esta introducción a la CPU ha sido de carácter muy general; los puntos específicos se desarrollarán en detalle más adelante. El propósito ha sido mostrar que las instrucciones especiales leídas en la memoria hacen que la CPU lleve a cabo determinadas operaciones y disponga las patillas de dirección para poder acceder a puntos concretos de la memoria. Los datos son traídos desde estos puntos, o enviados a ellos, mediante el bus de datos. Las órdenes hacen, asimismo, que las señales del bus de control indiquen a la memoria o al I/O si estos datos deben ser "leídos" o "grabados".



John von Neumann

Este brillante matemático aporta su nombre al diseño de los ordenadores modernos



Associated Press

Pareja complementaria
John von Neumann junto a su segunda esposa, Klara, también una experta programadora de los primeros ordenadores

Sólo un húngaro podría entrar en una puerta giratoria tras una persona y salir delante de ella. Esto dijo John von Neumann al describir la competitividad de los colegas de su propio país.

Él mismo no fue una excepción. Su propia ambición, junto con una inteligencia extraordinaria, le condujo a los más altos puestos científicos de los Estados Unidos.

Neumann nació en el seno de una acaudalada familia judía del Imperio austrohúngaro. Su facilidad para las matemáticas ya se puso de manifiesto cuando aún era muy joven, y a los 25 años de edad había obtenido dos licenciaturas y un doctorado y discutía temas científicos en un plano de igualdad con personajes tan eminentes como Albert Einstein y el matemático David Hilbert.

Neumann nunca fue indiferente a los problemas del mundo. Con el derrumbamiento del Imperio austrohúngaro, tras la primera guerra mundial, adoptó la partícula nobiliaria *von* y se introdujo en la vida académica de la derrotada Alemania. Al mismo tiempo, establecía contactos en los Estados Unidos, pasando los inviernos en la Universidad de Princeton, en Nueva Jersey, y los veranos en Europa, administrando las propiedades de su padre.

Al estallar la segunda guerra mundial, ya se había establecido en Norteamérica.

El prestigio de Von Neumann en el campo de las matemáticas se debió a sus estudios sobre la teoría de conjuntos, que Bertrand Russell había desprestigiado con sus paradojas lógicas. Von Neumann estaba fascinado con la física cuántica y con la teoría de los juegos. Inventó el método Montecarlo, en el que se utilizan números aleatorios para resolver ecuaciones matemáticas.

Cuando el gobierno de Estados Unidos decidió entrar en la guerra al lado de los Aliados, Von Neumann fue contratado inmediatamente para el proyecto Manhattan, y colaboró con gran entusiasmo en la fabricación de la bomba atómica. Durante su estancia en el centro de investigación de Los Álamos, viajaba con frecuencia cerca de 200 km en su coche para comer en su restaurante mexicano favorito, y durante sus últimos años en Princeton se dice que destrozaba un automóvil al año por su desastrosa forma de conducir.

Cuando aún estaba comprometido en el proyecto Manhattan, tuvo noticias de los intentos de construir un ordenador electrónico y solicitó participar en el proyecto ENIAC. Los trabajos se realizaban bajo el control de ingenieros electrónicos, pero como primer matemático involucrado vio el problema de forma diferente y realizó un informe que se convertiría en punto de partida del ordenador moderno.

Finalizada la guerra, colaboró cada vez más estrechamente con el Departamento de Defensa de Estados Unidos. No abandonó, sin embargo, sus investigaciones matemáticas y diseñó el primer ordenador para la Universidad de Princeton, que recibió el nombre de JOHNIAC. En una fiesta para celebrar la conclusión del ordenador, Von Neumann recibió una maqueta del mismo esculpida en hielo.

Con algo más de 50 años, se le declaró un cáncer y posteriormente se vería confinado a una silla de ruedas.

A pesar de que toda su vida fue un agnóstico, en los últimos meses de su existencia se convirtió a la religión católica.

A su muerte, un amigo suyo dijo: "Nunca había visto sufrir a un ser humano tanto como a Von Neumann cuando su mente ya no le respondía".

A partir de entonces, y en su honor, el diseño de ordenadores es conocido como *arquitectura Von Neumann*.

1903
El 28 de diciembre nace en Budapest Janos Louis Neumann

1921
Primer escrito matemático publicado junto con su tutor

1933
Es nombrado profesor de la Universidad de Princeton (Nueva Jersey, Estados Unidos), refugio de Einstein y de otros matemáticos emigrados de Europa

1942
Publica un libro sobre la aplicación de la teoría de juegos a la economía

1944
Tiene noticias sobre el proyecto secreto ENIAC

1945
Asiste a las primeras pruebas de la bomba atómica

1947
Informe sobre el ENIAC y bosquejo del diseño de un nuevo ordenador, que recibirá el nombre de EDVAC (Electronic Discrete Variable Computer) y será construido según sus ideas

1951
JOHNIAC, el ordenador diseñado personalmente por Neumann, entra en funcionamiento en la Universidad de Princeton

1951-1953
Presidente de la American Mathematical Society. Trabaja en teoría de autómatas

1955
El presidente Eisenhower le nombra comisionado de la US Atomic Energy

1957
El 8 de febrero muere de cáncer. Como un honor especial se dedica un volumen completo del *Bulletin of the American Mathematical Society* a su vida y obra

"La voz de su amo"

Los ordenadores ya ocupan un lugar propio en el campo de la música profesional. También se están incorporando pequeños sintetizadores de música en muchos ordenadores personales

Los ordenadores son a la vez divertidos y serios. Y no sirven tan sólo para procesar información o para jugar. También se pueden utilizar como instrumentos musicales para, por decirlo de alguna manera, divertirse con seriedad. El proceso de hacer música artificialmente se denomina síntesis musical.

Los ordenadores también se pueden utilizar para hacer más amena la enseñanza de la música y su concurso resulta bastante más económico que contratar a un profesor particular. En un próximo artículo le explicaremos cómo realizar ese tipo de música con un ordenador personal, pero de momento nos concentraremos en cómo lo hacen los profesionales del medio.

Los instrumentos musicales automáticos siempre han gozado de popularidad y tienen mucho en común con los ordenadores. La pianola, una especie de piano automático que durante el siglo pasado solía adornar las salas de estar de las familias acomodadas, funcionaba mediante un rollo de papel perforado, y las cajas de música llevaban un disco o un tambor metálico con "dientes" que ejecutaban una melodía sobre un peine de metal.

En cierto sentido, hasta los organillos callejeros que funcionaban a manivela eran programables, ya que las melodías que ejecutaban se podían sustituir por otras. No obstante, estos organillos no eran del agrado de Charles Babbage, uno de los padres fundadores de la informática, quien era partidario de que a los organilleros se les prohibiera hacer música en las calles. Éstos, como respuesta, acudían a tocarle melodías bajo su propia ventana.

En la actualidad, en Gran Bretaña, es el Sindicato de Músicos el que está tratando de impedir el uso de dispositivos musicales programables; en mayo de 1982, la sección sindical central de Londres decidió, mediante votación, prohibir su utilización en las sesiones de grabación y en las actuaciones en vivo. Como es evidente, esta preocupación de los sindicatos de músicos nace del hecho de que, dado que estos dispositivos pueden imitar el sonido de muchos instrumentos diferentes y de forma simultánea, con el tiempo los músicos no serían necesarios.

Los sintetizadores electrónicos salieron al mercado hace ya muchos años, pero la introducción de técnicas digitales les ha abierto un campo totalmente nuevo. En vez de tener que manipular clavijas y pulsar botones para producir cada uno de los sonidos, ahora se puede grabar cualquier sonido, analizarlo mediante ordenador en las partes que lo constituyen y reproducirlo en cualquier tono.

El sonido digitalizado es algo así como una fotografía publicada en un periódico; si mira la página desde muy cerca, verá que la imagen se compone de muchos puntos pequeños y separados unos de otros, mientras que la fotografía original (analógica) tiene tonalidades difuminadas de sombras que se mezclan continuamente entre sí. Del mismo modo, el sonido analógico normal se puede descomponer en una secuencia de dígitos. Esta técnica se denomina muestreo.



Cortesía de EMI

Un hombre, una orquesta
Los músicos sintetizadores como Klaus Schultz, a quien vemos en la fotografía, están utilizando cada vez más el enorme potencial de sus instrumentos basados en microprocesadores para producir, en vivo, la misma variedad de sonidos para la que hace veinte años se hubiera necesitado toda una orquesta

Los sistemas de este tipo son caros (probablemente los modelos más conocidos y menos sofisticados sean el Fairlight y el Synclavier), pero, puesto que pueden reproducir los sonidos de varios instrumentos musicales, resultan más baratos que contratar a todos los músicos que serían necesarios para este fin.

Con el descenso de los precios de los ordenadores y la progresiva disminución del costo de la memoria, está aumentando la popularidad de las máquinas digitales, aunque falta aún mucho tiempo para que los sintetizadores analógicos desaparezcan por completo. Estos últimos utilizan una técnica denominada "síntesis de sustracción" que, de alguna forma, puede compararse con la manera en que un escultor moldea su estatua en un bloque de mármol. Se comienza con un sonido básico creado electrónicamente y luego se lo hace pasar a través de una serie de procesos electrónicos. Cada proceso modifica o descompone el sonido

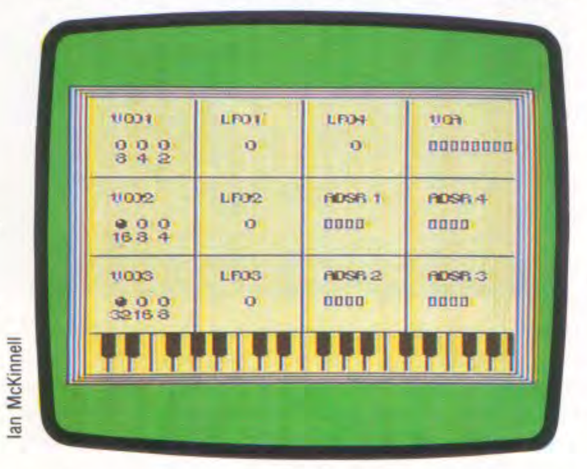


Ian McKinnell

Efectos especiales
Los sintetizadores de música controlados por ordenadores personales se están haciendo cada vez más populares. Los efectos musicales, que hace diez años sólo existían para los equipos profesionales más costosos, ahora tienen un precio muy asequible. El que vemos en la fotografía puede leer en su memoria música que previamente ha sido codificada en papel en forma de código de barras. Luego el ejecutante la puede reproducir o modificar. Muchos de estos sintetizadores personales se pueden acoplar directamente a un ordenador personal, para aprovechar la pantalla o la memoria adicional. Sin embargo, cada vez son más numerosos los ordenadores personales que llevan incorporada alguna forma de síntesis musical

en sus mínimos componentes, para darle la forma deseada. La síntesis de sustracción estimula a experimentar con diferentes combinaciones de procesos y su técnica resulta sencilla hasta para un principiante.

Por el contrario, todo cuanto se cree con un sintetizador digital ha de planificarse tan cuidadosamente como la construcción de un gran edificio. Ello se debe a que el dispositivo se vale de la *síntesis de adición*: el sonido final se produce mediante la adición de componentes, uno sobre el otro. Uno ha de estar próximo al final del proceso para poder ser capaz siquiera de reconocer mínimamente el sonido. Sin embargo, se puede tomar un sonido convencional, descomponerlo en sus componentes básicos, almacenar éstos en la memoria del ordenador, ya sea en una RAM interna o en



Ian McKinnell

Cajas de música
Actualmente existen muchos paquetes para ordenadores personales que permiten aprovechar al máximo la capacidad que éstos poseen para realizar música. En estos paquetes, la visualización en pantalla se puede utilizar para hacer una intepretación visual de la música que se está ejecutando, o para ayudar al músico recién iniciado, que puede pulsar el teclado QWERTY como si fuera el de un piano

un disco o una cinta, y utilizarlos como "ladrillos" para construir el sonido deseado.

Casi tan importante como su aptitud para crear una variedad tan amplia de sonidos individuales es la capacidad que poseen los ordenadores para almacenar secuencias y composiciones musicales. La mayoría de los sintetizadores más vendidos poseen un secuenciador (dispositivo que almacena y recupera las secuencias de sonidos) opcional, aunque es probable que ya lo lleven incorporado. El Fairlight, por ejemplo, puede almacenar en su memoria de disco 30 minutos de sonido de hasta ocho "voces", que se pueden considerar como instrumentos musicales. La capacidad del Synclavier corresponde al doble.

De todo lo anterior se deduce que en los sintetizadores más caros los papeles tradicionalmente asignados al compositor, los músicos y el director se funden en el de una sola persona, que, en gran medida, es, además, un programador.

Si alguna vez ha intentado reproducir a una velocidad mayor una cinta grabada con su voz (o si con ese fin ha reproducido a 45 r.p.m. un disco de 33 r.p.m.), se habrá dado cuenta de que el tono se eleva considerablemente. Una de las características más sorprendentes de los sintetizadores controlados por ordenador es su capacidad para salvar este inconveniente y reproducir una pieza musical a mayor o menor velocidad que la original sin alterar su tono, o, por el contrario, trasponer la melodía a una clave distinta a la misma velocidad.

Incluso es posible tomar, por ejemplo, la parte de una trompeta, duplicarla y, simultáneamente, modificar el sonido para que adopte el de una trompa francesa. Es factible, entonces, tocar ambos instrumentos armónicamente, sea al unísono o de forma alternada Esto se conoce con el nombre de *track bouncing*.

A muchas de estas máquinas se les proporcionan las instrucciones por medio de un lenguaje de composición (el del Synclavier se denomina SCRIPT) que no difiere demasiado del BASIC (también posee números de líneas), aunque tal vez sea un poco más difícil de utilizar. La configuración más relevante es la denominada *compilación invertida*: se ejecuta una pieza musical en el teclado y el ordenador produce un listado en SCRIPT para su composición. Esto equivaldría, si es que uno se pudiera imaginar algo así, a que se pudiera jugar a un juego nuevo creado por uno mismo en la pantalla del ordenador y luego, apretando un botón, ¡se obtuviera un listado completo del programa para este juego!

Si considera que su sentido del ritmo no ha sido todo lo perfecto que deseaba, existe la posibilidad de editar y volver a reproducir el listado en SCRIPT utilizando un teclado convencional (QWERTY) y una pantalla, igual que en BASIC. Uno de los sintetizadores Yamaha tiene un sistema menos flexible pero más atrayente: la melodía que usted haya ejecutado la imprime utilizando el procedimiento musical convencional, es decir, por medio de notas en el pentagrama.

Los sintetizadores también se están introduciendo en el cine. La película *TRON*, producida por Walt

Disney, ha recibido entusiastas críticas, que elogian sus sorprendentes gráficos realizados por ordenador. Mucho menos conocido resulta el hecho de que se emplearon microordenadores para la música y los efectos sonoros. Además de la parte de música "verdadera" ejecutada por la London Philharmonic Orchestra, se grabó y se modificó una gran variedad de sonidos usando un sintetizador Fairlight, incluyendo los ruidos producidos por la aeronave Goodyear y por una nevera.

Fue tal la cantidad de sonidos que se emplearon, que se hubo de mantener actualizado un catálogo completo mediante el paquete de base de datos File Manager 800+, procesado con un ordenador personal Atari 800. Algunas de las voces artificiales de la película también se realizaron mediante un ordenador personal y un dispositivo para síntesis de voz que permitía mezclar voces y música. Atari también les facilitó un paquete que jamás anteriormente se había utilizado fuera de dicha empresa. Consistía en una clase de programa para síntesis de adición que Atari había venido empleando para crear sofisticados efectos de sonido en sus propios programas para juegos y para sus máquinas tragaperras. La impresión general es que la importancia de este "sistema electrónico ensamblador de sonidos" respecto a la creación de sonidos es equiparable a la que tuvo el tratamiento de textos en relación a la creación de textos.

En realidad existe un número bastante elevado de accesorios con los cuales se puede convertir un ordenador personal en un sofisticado sintetizador de música. El Apple es un modelo especialmente popular para este tipo de aplicación, debido a las ranuras para conectar dispositivos adicionales que posee el ordenador en su parte posterior. La finalidad de estos dispositivos es hacer que el ordenador realice todo el trabajo creativo y proporcionar un sintetizador analógico de gran calidad como salida. Otros incluyen un teclado exactamente igual al de un piano.

En el extremo más doméstico del mercado de sintetizadores informatizados, el Casio CT7000 incluye lo que se conoce como *secuenciador polifónico*, el cual, mediante una unidad de cassette y una gran cantidad de memoria RAM, puede funcionar como un sistema de grabación profesional de pistas múltiples. Una grabadora de cassette corriente sólo posee una pista por cada lado, mientras que una estereofónica posee dos. Pero una unidad profesional puede llegar a tener más de 24, de modo que la interpretación de cada instrumento se puede grabar por separado, para luego mezclar todos los sonidos para obtener la banda sonora definitiva. De esta manera, con el CT7000 uno puede crear sus propias sinfonías.

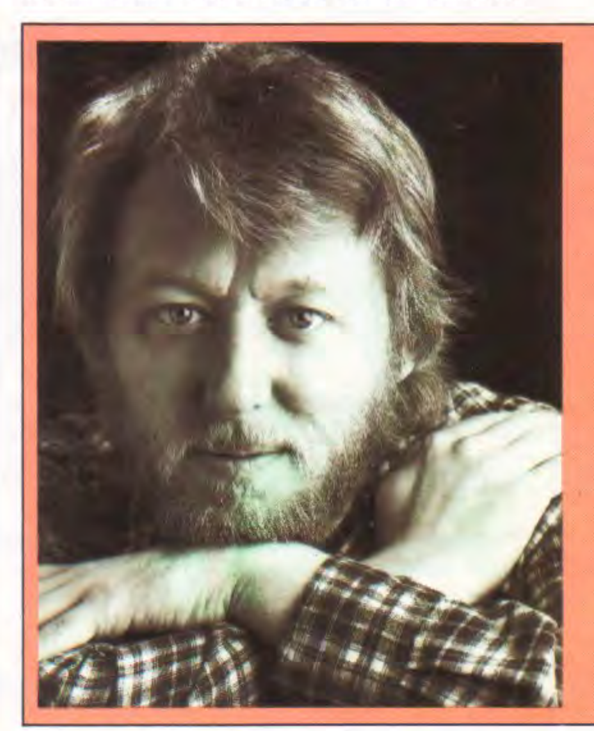
Su antecesor, el CT701, utilizaba un lector de código de barras (véase p. 40) para leer música a partir de los códigos de barras impresos en la memoria de la máquina. Lamentablemente, parecería que no existen medios viables para que el usuario pueda crear códigos de barras impresos de sus propias creaciones, por lo cual este procedimiento sólo se puede emplear con la música preimpresa del sintetizador Casio.

Para triunfar en el mundo de la música pop, ni siquiera se necesita contar con un equipo de este nivel, como lo ha demostrado el grupo alemán Trio, que con su tema *Da Da Da*, para cuya grabación utilizaron un Casiotone VL1 que cuesta 30 libras (unas 6 700 pesetas), obtuvieron más éxito que el que Peter Gabriel consiguiera con un Fairlight de miles de libras.

A pesar de que el VL1 es un dispositivo monofónico (en el cual sólo se puede ejecutar una nota a la vez), es capaz de emular diversos instrumentos musicales y de almacenar una secuencia de notas. También ofrece la posibilidad de cambiar determinadas cualidades de las notas para que el usuario tenga la posibilidad de crear su propio sonido. Esto se conoce como alterar la envoltura ADSR (Attack, Decay, Sustain, Release: atacar, decaer, sostener, soltar). Ahora bien, lo más interesante es que ordenadores personales como el Commodore 64 y el Oric-1 poseen exactamente las mismas características. Y, por supuesto, el solo hecho de que el aparato sea programable en BASIC le da la facultad de contar con un secuenciador incorporado.

Existen en el mercado paquetes de programas para composición musical para muchos de estos ordenadores personales, incluso para los que, comparativamente, poseen configuraciones musicales más sencillas. Algunos de ellos visualizan en pantalla un pentagrama, y la música se compone seleccionando los diferentes tipos de notas musicales con la ayuda de una palanca de mando o de un lápiz óptico, y colocándolas en el pentagrama. Al pulsar el botón de "disparo" de la palanca de mando, o dar alguna orden igualmente sencilla, la composición musical empieza a ser ejecutada. Otra alternativa consiste en que la visualización represente el teclado de un piano; en este caso las notas también se seleccionan con el lápiz óptico o con la palanca de mando, o bien con el teclado del ordenador.

Los efectos musicales pueden programarse en BASIC sin que sea necesario utilizar un paquete de este tipo. Al igual que las configuraciones para gráficos, el procedimiento exacto varía considerablemente de una máquina a otra, así como la sofisticación de las órdenes en BASIC creadas con este fin. El Dragon, por ejemplo, posee una sola voz, pero dispone de una orden PLAY que inicia una secuencia de notas, que se digitan como la antigua notación de la escala musical: de la A a la G (de *la* a *sol*, según la notación actual). El Commodore 64, por el contrario, tiene incorporadas en su hardware muchas y sofisticadas configuraciones musicales, pero no posee órdenes en BASIC listas para utilizar, como el Dragon. Próximamente le enseñaremos cómo realizar música con un ordenador.



Martin Rushent

Productor de grupos como Altered Images y Dexy's Midnight Runners, Martin Rushent es uno de los vanguardistas de los sintetizadores musicales controlados por ordenador. En su estudio del distrito rural de Berkshire hay no menos de nueve sistemas diferentes, que representan la situación actual de este arte. Cada uno de estos sistemas utiliza un procedimiento de programación diferente y distintos códigos para las notas de una escala musical. Puesto que no ha sido factible conectarlos a todos entre sí para formar un único sistema gigante, Rushent, para trasponer las piezas musicales de uno a otro sistema, utiliza un programa que escribió él mismo en un pequeño ordenador personal.

Len Lewis

Bien direccionada

La CPU ha de localizar las instrucciones y los datos almacenados en miles de bytes de memoria del ordenador. Le revelamos lo que sucede en la CPU al ejecutar las instrucciones de un programa

La CPU recibe sus instrucciones y sus datos desde las direcciones situadas en la memoria del ordenador, colocando las patillas de dirección en el código binario requerido por la localización de memoria y leyendo luego el contenido de la localización en la CPU a través del bus de datos. Expuesto de esta manera, el proceso no parece ofrecer mayores dificultades; sin embargo, nada más lejos de la realidad, por cuanto en la práctica esta operación es bastante más complicada.

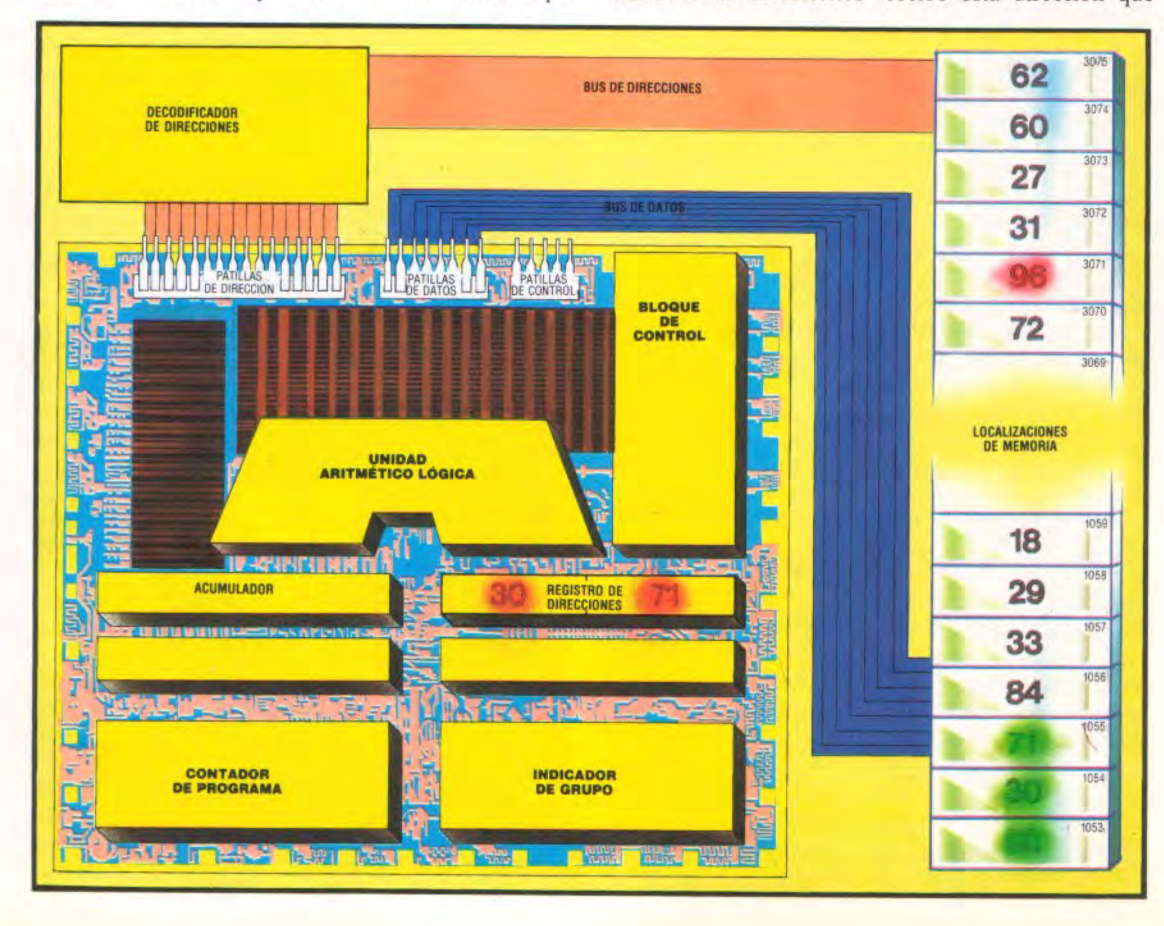
El problema reside en que los bytes (códigos binarios de ocho bits) de cualquiera de las miles de celdas de memoria del ordenador pueden ser instrucciones que le digan a la CPU que realice algo, o bien información que la CPU ha de manipular de alguna forma. ¿Cómo sabe la CPU cuáles bytes son instrucciones y cuáles son datos?

Reconociendo los códigos

En primer lugar, veamos qué es una "instrucción". Una instrucción es un código, en binario, que hace que en el interior de la CPU se lleve a cabo una secuencia específica de operaciones. De modo que el código 00111010, en caso de que la CPU lo reconozca como una instrucción y no tan sólo como un dato, podría hacer que la CPU direccionara los dos próximos bytes de la memoria, leyera los datos que contienen, colocara esos datos en un "registro de direcciones" especial, situara las patillas de dirección en el mismo número, fuera hasta la localización de memoria recientemente direccionada, a continuación colocara el contenido de dicha localización en el bus de datos y, para finalizar, cargara aquellos contenidos en el acumulador de la CPU.

Todo esto, expresado en palabras, puede resultar confuso; pero lo que acabamos de describir es uno de los procedimientos de direccionamiento de memoria que utiliza la CPU del popular Z80. El gráfico ilustra el proceso completo de tomar un byte de información de la memoria y colocarlo en la CPU. Supongamos que la CPU ya sabe que el próximo byte de la memoria al cual accederá será una instrucción (y no información) y que este byte está en la localización de memoria 1053. (Todos los números utilizados en esta ilustración están en notación decimal.) Esta dirección, la 1053, se colocará en el bus de direcciones. En binario es 0000010000011101. Las 16 patillas de dirección están en posición "encendido" o "apagado" de modo tal que correspondan a este número. Cuando el "decodificador de direcciones" recibe esta dirección que

Cadenas de acontecimientos
Aun la operación más sencilla de la CPU implica muchas etapas. Las instrucciones, también llamadas "códigos de operación", se le leen a la CPU desde la memoria. Estas instrucciones son decodificadas por el bloque de control y hacen que se lleven a cabo operaciones específicas. En este ejemplo, se lee la instrucción 58 de la localización de memoria 1053. Esta instrucción determinada hace que se produzca la siguiente cadena de acontecimientos: el byte de la siguiente localización de memoria (1054) será leído y almacenado en una mitad del registro de direcciones de 16 bits de la CPU. El byte de la localización siguiente (1055) será leído y almacenado en la otra mitad. Ahora estos dos bytes representan la dirección (en algún otro lugar de la memoria) donde se han almacenado unos datos. El contenido del registro de direcciones se coloca en el bus de direcciones, de modo que la próxima localización de memoria a la que se acceda será la dirección 3071. El contenido de esta dirección se pone en el bus de datos y se le lee a la CPU. Este byte (el 96 en nuestro ejemplo) es colocado entonces en el acumulador de la CPU, donde permanecerá hasta que una instrucción posterior lo requiera. El bus de direcciones regresará entonces a su anterior dirección +1, de manera que ahora se dirigirá a la localización 1056. La CPU sabe que, sea lo que fuere lo que contenga esa localización, ha de ser una instrucción, y que se repetirá una secuencia de operaciones similares. En este ejemplo la siguiente instrucción es la 84, que es interpretada por el bloque de control para "complementar" o invertir los bits del acumulador. Dado que 84 es una instrucción "de un byte", la CPU sabe que el byte de la próxima localización de memoria, la 1057, también será una instrucción



viene en el bus de direcciones, la "decodifica" y enciende una, y sólo una, de sus líneas de salida. Ésta es la línea que se encarga de seleccionar la localización de memoria 1053.

La próxima etapa consiste en colocar el contenido de esta dirección, que es 58, o 00111010 en binario, en el bus de datos y "cargarlo" en la CPU. Aquí, como la CPU está esperando una instrucción, el byte es interpretado por el bloque de control y hace que se realice una secuencia muy precisa de operaciones. Esta instrucción en particular especifica que los dos próximos bytes de memoria contendrán 16 bits para ser utilizados como localización de memoria, y que el contenido de esta localización se ha de cargar en el acumulador de la CPU. No bien la CPU reconoce esta instrucción, sabe que los dos próximos bytes de memoria especificarán una dirección y que el contenido de esa dirección se habrá de cargar en el acumulador. En consecuencia, sabe que no recibirá otra instrucción desde la memoria hasta después que se hayan efectuado estas operaciones, y que la próxima instrucción estará en la localización 1056.

La instrucción que estamos utilizando en el ejemplo hace que el bus de direcciones se incremente en 1, de manera que la próxima localización de memoria que se direcciona es la 1054. El contenido de esta localización se coloca entonces en el bus de datos y se carga en la CPU. Esta vez, sin embargo, se coloca en la mitad de un registro de dirección. Después de hacerlo, la CPU vuelve a incrementar el bus de direcciones de modo que ahora direccione la ubicación 1055. El contenido de esta localización se coloca en el bus de datos e, igualmente, se carga en la CPU, si bien en esta ocasión pasa a almacenarse en la otra mitad del registro de direcciones.

Transfiriendo números

La próxima etapa (recuerde que todas estas acciones se producen automáticamente a consecuencia de la instrucción original) consiste en que los números del registro de direcciones se transfieren al bus de direcciones. Estos números, como podemos ver, son 3071. Por lo tanto, la localización de memoria que se está direccionando ahora es la 3071. Esta dirección (en binario 0000101111111111) es decodificada por el decodificador de direcciones y selecciona la celda de memoria 3071. El contenido de esta localización, 96 (en binario 01100000), se coloca en el bus de datos y se carga en la CPU. Esta vez, sin embargo, la información se colocará en el acumulador de la CPU. Después de efectuada esta operación, el bus de direcciones se establecerá en 1056 y allí la CPU esperará encontrar otra instrucción.

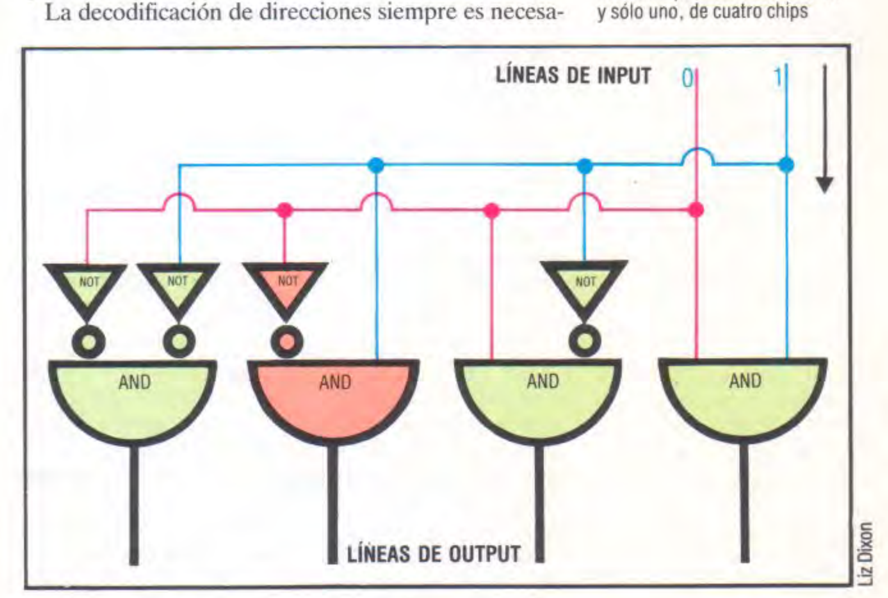
Ahora que la CPU posee en su acumulador una información determinada, ¿qué clase de instrucción se podría esperar a continuación? Podría ser prácticamente cualquiera; las CPU son capaces de reconocer desde docenas hasta centenares de instrucciones, según la CPU. Pero supongamos que lo que deseábamos era invertir los datos del acumulador. Invertir significa cambiar cada uno por un cero y cada cero por un uno. La instrucción para hacerlo estaría localizada en la dirección 1056. En nuestra CPU imaginaria, el código para esta instrucción sería 84. Cuando la CPU recibiera este número, se invertirían los datos del acumulador. El número que estaba en el acumulador era 96 (01!00000 en binario). La instrucción para invertir haría que éste se sustituyera por el número binario 10011111. La instrucción para invertir un número del acumulador es una instrucción "de un byte", de modo, pues, que nuevamente la CPU sabría que el contenido de la siguiente localización de memoria, 1057, será otra vez una instrucción y no un dato.

Este procedimiento de direccionar una localización de memoria para recuperar un dato sólo es uno de los diversos métodos de que dispone el programador. Los bytes de instrucción específicos que hemos utilizado en el ejemplo (58 para cargar el acumulador y 84 para invertir los contenidos del acumulador) son instrucciones para nuestra CPU hipotética, pero el mismo principio se aplica para todos los otros chips de microprocesadores. La única diferencia estriba en que se emplean códigos distintos para las diversas instrucciones y que cada versión de CPU posee su propio "juego de instrucciones", que difiere ligeramente del de las demás.

Las localizaciones de I/O (Input/Output) también deben poseer direcciones exclusivas, pero los principios en virtud de los cuales la CPU las direcciona son los mismos. Por lo general, en los microprocesadores de ocho bits sólo ocho de las líneas de dirección están disponibles para el direccionamiento de I/O, de manera que el número máximo de direcciones de I/O es 256. No obstante, esto es más que suficiente para la mayoría de las aplicaciones de los ordenadores pequeños.

La decodificación de direcciones siempre es necesaria, de manera que el dispositivo seleccionado por la CPU (ya sea una localización de memoria o una localización de I/O) está especialmente activo cuando todas las demás localizaciones de memoria o de I/O están inactivas. Este proceso se denomina *activación*. Cuando hay que decodificar una pequeña cantidad de líneas de dirección, se pueden utilizar chips de puertas lógicas simples para que realicen la decodificación. El gráfico ilustra el principio del decodificador de línea de dos a cuatro. Este tipo de decodificación sencilla se emplea con frecuencia para seleccionar dispositivos de I/O. Sin embargo, a medida que va aumentando el número de líneas de dirección, la complejidad del circuito de decodificación se incrementa de manera considerable. Cuando es preciso proceder a seleccionar individual y exclusivamente 65 536 localizaciones de memoria separadas, lo normal es que la mayor parte de la decodificación de direcciones se lleve a cabo en el interior de los chips de memoria.

Decodificación de direcciones
Las 16 líneas que constituyen el bus de direcciones son capaces de identificar de modo exclusivo cualquiera de las 65 636 localizaciones de memoria separadas. La combinación de unos y ceros del bus de direcciones se decodifica en los decodificadores de direcciones. Parte de la decodificación la efectúan los decodificadores de direcciones, compuestos por puertas lógicas simples en chips montados en el circuito; gran parte de la decodificación la efectúan circuitos equivalentes situados dentro de los propios chips de memoria. La ilustración muestra cómo se pueden decodificar dos líneas de dirección para seleccionar uno, y sólo uno, de cuatro chips



Totalmente funcional

El BASIC posee funciones incorporadas; ello significa que gran parte de la programación ya ha sido hecha para usted. El saber utilizarlas le permitirá disponer de un mayor potencial informático

Supongamos que en uno de sus programas desea averiguar la raíz cuadrada de un número. La puede calcular de diversas maneras. La forma más elemental y menos satisfactoria sería crear una tabla de valores de raíz cuadrada y utilizarla para obtener el valor deseado para un número determinado. Es probable que haya aprendido en la escuela la forma de confeccionar una tabla de este tipo. Un método alternativo consiste en utilizar la "función" raíz cuadrada, que incorporan la mayoría de las versiones de BASIC. Aquí es este lenguaje el que se encarga de la aritmética de la operación, sin que el programador deba preocuparse por ella. Veamos cómo funciona:

```
10 REM ESTE PROGRAMA BUSCA LA RAIZ
   CUADRADA
20 REM DE UN NUMERO
30 PRINT "DE ENTRADA AL NUMERO CUYA RAIZ
   CUADRADA"
40 PRINT "DESEA HALLAR"
50 INPUT N
60 LET A = SQR(N)
70 PRINT "LA RAIZ CUADRADA DE"; N; "ES"; A
80 END
```

Digite este corto programa y verá que, en efecto, le proporcionará la raíz cuadrada de cualquier número que digite. Estudiemos las reglas relativas a la forma de utilizar esta función de "raíz cuadrada".

Una "función" en BASIC es generalmente una palabra de orden (en este caso SQR, abreviatura de *square root*: raíz cuadrada) seguida de paréntesis que encierran la expresión a operar. En este programa, N es la entrada del número desde el teclado. Es el número cuya raíz cuadrada deseamos hallar. La línea 60 dice "asignemos la raíz cuadrada de N a la variable A". La línea 70 imprime el valor de A.

La expresión entre paréntesis se denomina *argumento* de la función y no siempre ha de ser una variable: es igualmente posible utilizar números reales. Digite lo siguiente y vea qué sucede cuando lo ejecuta:

```
10 PRINT SQR(25)
20 END
```

Comprobará que esto funciona exactamente igual. De la misma manera, podemos incluir entre los paréntesis argumentos más complicados. Pruebe con este ejemplo:

```
10 LET A = 10
20 LET B = 90
30 LET C = SQR(A + B)
40 PRINT C
50 END
```

Este pequeño programa se puede acortar combinando las líneas 30 y 40 de la siguiente manera:

```
10 LET A = 10
20 LET B = 90
30 PRINT SQR(A + B)
40 END
```

Las funciones se deben considerar como programas cortos incorporados al BASIC, disponibles para que el programador los emplee en cualquier momento. La mayoría de las versiones de BASIC ofrecen una cantidad bastante importante de funciones, así como la posibilidad de que el programador defina algunas nuevas para utilizarlas dentro de un programa. Más adelante veremos la forma de hacerlo. Ahora estudiaremos algunas de las otras funciones disponibles comúnmente. Éstas presentan dos variedades: las funciones numéricas, en las cuales el argumento (la parte encerrada entre paréntesis) es un número, una variable numérica o una expresión numérica, y las funciones en serie, en las cuales el argumento es una serie de caracteres o una expresión alfanumérica. Veamos primero algunas de las funciones numéricas.

Con anterioridad, en la página 77, ejecutamos un programa que calculaba el número de azulejos que se necesitaban para revestir las paredes de una habitación. Un pequeño *bug* de este programa consistía en que la respuesta no podía incluir fracciones decimales de un azulejo. La ejecución de este programa podría dar 988,24 como posible resultado. En ocasiones como ésta deseamos disponer de algún procedimiento para redondear la respuesta en el número completo más próximo. En matemáticas, los números completos se denominan *enteros* y una de las funciones del BASIC es "devolver" la parte entera de cualquier número. Funciona así:

```
10 PRINT "DE ENTRADA A UN NUMERO QUE
   CONTENGA FRACCION DECIMAL"
20 INPUT N
30 PRINT "LA PARTE ENTERA DEL NUMERO ES";
40 PRINT INT(N)
50 END
```

Si ejecuta este programa y da entrada a 3,14, el programa imprimirá en pantalla:

```
LA PARTE ENTERA DEL NUMERO ES 3
```

Por supuesto, si se trata de azulejos necesitaríamos luego sumarle 1 a la respuesta, para asegurarnos de comprar más cantidad de la requerida y no menos.

En otra ocasión podríamos desear averiguar el "signo" de un número para ver si es negativo, cero o positivo. Para ello, la mayoría de las versiones de BASIC incorporan una función SGN. Ensayemos con el siguiente ejemplo:

```
10 PRINT "DE ENTRADA A UN NUMERO"
20 INPUT N
```

```
30 LET S = SGN(N)
40 IF S = -1 THEN GOTO 100
50 IF S = 0 THEN GOTO 120
60 IF S = 1 THEN GOTO 140
100 PRINT "EL NUMERO ERA NEGATIVO"
110 GOTO 999
120 PRINT "EL NUMERO ERA CERO"
130 GOTO 999
140 PRINT "EL NUMERO ERA POSITIVO"
150 GOTO 999
999 END
```

Si observa los valores "devueltos" a S en la línea 30 por la función SGN (éstos se comparan en las líneas 40, 50 y 60), verá que los valores son tres. Se devuelve −1 si el argumento entre paréntesis era un número negativo, 0 si el argumento era cero y 1 si el argumento era un número positivo. La utilización de la función SGN en la línea 30 ahorra varias líneas de programación. Podríamos haber escrito:

```
IF N < 0 THEN LET S = -1
IF N = 0 THEN LET S = 0
IF N > 0 THEN LET S = 1
```

La acción que se consigue realizar mediante el empleo de una función de BASIC siempre se puede obtener mediante la programación normal, pero el empleo de una función supone un ahorro de tiempo, de espacio y de esfuerzo de programación.

A continuación reseñamos unas pocas funciones numéricas más. ABS devuelve el valor "absoluto" de un número. El valor absoluto de un número es el de su valor real pero sin su signo. Por lo tanto, el valor absoluto de −6 es 6. Probemos:

```
10 LET X = -9
20 LET Y = ABS(X)
30 PRINT Y
40 END
```

MAX busca el valor máximo de dos números. Por ejemplo:

```
10 LET X = 9
20 LET Y = 7
30 LET Z = X MAX Y
40 PRINT Z
50 END
```

MIN es similar a MAX, pero averigua el valor más pequeño de dos números. Veamos el proceso:

```
10 PRINT "DE ENTRADA A UN NUMERO"
20 INPUT X
30 PRINT "DE ENTRADA A OTRO NUMERO"
40 INPUT Y
50 LET Z = X MIN Y
60 PRINT Z
70 END
```

Observe que estas dos últimas funciones poseen dos argumentos en lugar de uno, y que no es necesario encerrarlos entre paréntesis. La mayoría de las versiones de BASIC poseen también algunas otras funciones numéricas, incluyendo LOG para encontrar el logaritmo de un número, TAN para descubrir la tangente, COS para encontrar el coseno y SIN para hallar el seno. Más adelante veremos algunas de las maneras en que se pueden utilizar estas funciones "trigonométricas".

El BASIC posee, asimismo, varias funciones incorporadas que operan con variables alfanuméricas. En nuestro programa para seleccionar un nombre (p. 135) empleamos algunas de ellas, pero en aquella ocasión no estudiamos con detención la forma en que funcionaban. Ahora veremos con mayor detalle aquellas funciones en serie y algunas más.

Una de las funciones en serie más útiles es LEN. Ésta cuenta el número de caracteres de una serie de ellos encerrada entre comillas dobles o el número de caracteres asignados a una variable alfanumérica. Probemos con el siguiente ejemplo:

```
10 LET A$ = "COMPUTER"
20 LET N = LEN(A$)
30 PRINT "EL NUMERO DE CARACTERES DE LA
   VARIABLE ES"; N
40 END
```

¿Por qué necesitaríamos saber, alguna vez, de cuántos caracteres se compone una variable alfanumérica? Para ver por qué, dé entrada y ejecute este breve programa creado para construir un "triángulo de nombre". En primer lugar imprimirá la primera letra de una palabra, luego la primera y la segunda, después la primera, la segunda y la tercera, y así sucesivamente hasta imprimir la palabra completa.

```
5 REM IMPRIME UN 'TRIANGULO DE NOMBRE'
10 LET A$ = "JONES"
20 FOR L = 1 TO 5
30 LET B$ = LEFT$(A$,L)
40 PRINT B$
50 NEXT L
60 END
```

Ahora ejecute este programa. ¿Se imagina qué es lo que imprimirá? Se leerá algo similar a esto:

J
JO
JON
JONE
JONES

Este corto programa utiliza la función LEFT$ para extraer los caracteres de una variable. LEFT$ lleva dos argumentos. El primero especifica la variable y el segundo (que viene precedido por una coma) especifica el número de caracteres a extraer de la variable, empezando desde el lado izquierdo de ésta. A A$ se le había asignado la serie "JONES", de modo que LEFT$(A$,1) "devolvería" la letra J. LEFT$(A$,2) devolvería las letras JO. El breve programa anterior emplea un índice, L, que abarca desde 1 hasta 5, de manera que el segundo argumento de la función LEFT$ asciende desde 1 hasta 5 cada vez que se efectúa el bucle. Sabemos exactamente cuántos caracteres había en la palabra que deseábamos imprimir (JONES), por lo cual fue sencillo decidir que 5 sería el límite máximo del bucle FOR-NEXT. Pero ¿qué haríamos si no supiéramos de antemano cuántos caracteres habrá de tener el bucle?

Aquí entra en juego la función LEN. Esta función toma como argumento una variable (entre comillas dobles) o una variable alfanumérica. A continuación le proporcionamos algunos ejemplos para que vea cómo funciona LEN:

```
10 REM PROGRAMA PARA COMPROBAR LA
   FUNCION 'LEN'
20 PRINT LEN ("COMPUTER")
30 END
```

Al ser ejecutado, este programa imprimirá 8. Ha contado el número de caracteres que componen la palabra COMPUTER y ha devuelto este valor. Hagamos lo mismo pero de un modo ligeramente diferente:

```
10 REM BUSCANDO LA LONGITUD DE UNA VARIABLE
20 LET A$ = "MI COMPUTER"
30 LET L = LEN(A$)
40 PRINT L
50 END
```

Este programa, al ser ejecutado, habría de imprimir en pantalla 11. En esta variable hay 11 caracteres y no 10. Recuerde que, en lo que atañe al ordenador, el espacio que separa una palabra de otra es también un carácter. Ahora apliquemos la función LEN en una modificación de nuestro programa anterior para imprimir un "nombre triangular":

```
10 REM ESTE PROGRAMA IMPRIME UN 'TRIANGULO
   DE NOMBRE'
20 PRINT "DIGITE UN NOMBRE"
30 INPUT A$
40 LET N = LEN(A$)
50 FOR L = 1 TO N
60 LET B$ = LEFT$(A$,L)
70 PRINT B$
80 NEXT L
90 END
```

Cada vez que se ejecute este bucle, el valor de L se incrementará desde 1 hasta N (que es la longitud del nombre en la variable). Si ahora se da entrada al apellido SIMPSON, la línea 40 equivaldrá a LET N = LEN ("SIMPSON"), de manera que N se establecerá en 7. La primera vez que se efectúe el bucle, la línea 50 establecerá L en 1 y la línea 60 equivaldrá a LET B$ = LEFT$ ("SIMPSON", 1), de modo que a B$ se le asignará un carácter de la variable, empezando por la izquierda. Este carácter es S.

La segunda vez que se efectúe el bucle, L se establecerá en 2, de modo que la línea 60 equivaldrá a LET B$ = LEFT$ ("SIMPSON", 2). Ésta tomará los dos primeros caracteres de la variable y los asignará a la variable alfanumérica B$. Por lo tanto, B$ contendrá SI.

La función LEN descubrió que los caracteres de la variable SIMPSON eran 7 y le asignó este valor a la variable N, de manera que la última vez que se efectúe el bucle a B$ se le asignarán los 7 caracteres de la variable y se imprimirá la variable completa.

LEFT$ posee una función compañera, RIGHT$, que toma los caracteres de la variable alfanumérica exactamente de la misma forma pero comenzando por la derecha.

Por último, estudiaremos otra función alfanumérica, que también empleábamos en nuestro programa de clasificación de nombres. Se trata de INSTR y se utiliza para encontrar la localización de la primera aparición de una variable específica (denominada *subvariable*) dentro de una variable. En el programa de clasificación de nombres se utilizaba INSTR para localizar la posición del espacio entre el nombre de pila y el apellido. Funciona de la siguiente manera:

```
10 LET A$ = "GUARDABOSQUE"
20 LET P = INSTR(A$, "BOSQUE")
30 PRINT P
40 END
```

Antes de dar entrada a este programa y de ejecutarlo, intente predecir el valor que se imprimirá para P. Recuerde que INSTR localiza la posición inicial de la primera vez que se produce la subvariable dentro de la variable. Si la variable es GUARDABOSQUE, la posición inicial de la subvariable BOSQUE será 7, pues la B de BOSQUE es la séptima letra de GUARDABOSQUE. Algunas versiones de BASIC carecen de la función INSTR, pero poseen una función similar denominada INDEX. Ésta es la forma de utilizar INSTR (o INDEX) para localizar un espacio dentro de una variable:

```
10 REM BUSCANDO LA POSICION DE UN ESPACIO EN
   UNA VARIABLE
20 LET A$ = "ORDENADOR PERSONAL"
30 LET P = INSTR(A$, " ")
40 PRINT P
50 END
```

Observe que el segundo argumento de la función INSTR (línea 30) es " ". Las comillas encierran un espacio: el carácter que se ha de buscar. El programa imprimirá 10 como el valor de P, dado que el espacio se halla en el décimo lugar de la variable. Intente deducir qué se imprimiría si se modificara la línea 30 del siguiente modo:

```
LET P = INSTR(A$, "C")
```

Por último, una función muy práctica que se utiliza con la sentencia PRINT. Veamos lo que sucede al ejecutar este programa:

```
10 PRINT "ESTA LINEA NO ESTA SANGRADA"
20 PRINT TAB(5); "ESTA LINEA ESTA SANGRADA"
30 END
```

¿Puede ver lo que sucedió? La segunda línea se empezó a imprimir cinco espacios a la derecha del margen izquierdo. TAB es análoga al tabulador de una máquina de escribir. Aquí le ofrecemos otro breve programa que emplea la función TAB:

```
10 REM UTILIZANDO LA FUNCION TAB
20 PRINT "INTRODUZCA EL VALOR DE TAB"
30 INPUT T
40 LET W$ = "TABULACION"
50 PRINT TAB(T); W$
60 END
```

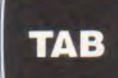
Ahora puede volver al programa de clasificación de nombres, en la página 136, y comprobar cómo se emplean en él algunas de estas funciones.

Ejercicios

■ **Bucles 1** ¿Qué se imprimirá al ejecutar este programa?

```
10 LET A = 500
20 FOR L = 1 TO 50
30 LET A = A -1
40 NEXT L
50 PRINT "EL VALOR DE A ES"; A
```

Complementos al BASIC



En el Spectrum reemplazar por el número de TAB.



El Spectrum no dispone de ninguna de estas órdenes, pero usted puede crear sus propias versiones de las mismas mediante la sentencia DEF FN; de modo que agregue a su programa las siguientes líneas:




```
9900 DEF FN L$(X$,N) = X$(TO N)
9910 DEF FN R$(X$,N) =
     = X$(LEN(X$) - N + 1 TO)
9920 DEF FN M$(X$,P,N) = X$(P TO
     P + N - 1)
```

Ahora



FN L$(X$,N) sustituye a LEFT$(X$,N)
FN M$(X$,P,N) sustituye a MID$(X$,P,N)
FN R$(X$,N) sustituye a RIGHT$(X$,N)



No disponen de esta función el Spectrum, el VIC-20, el Commodore 64 y el Oric-1, pero es posible sustituirla. Supongamos que la línea original fuera:

```
20 LET P = INSTR(A$,"BOSQUE")
```

Reemplácela por

```
20 LET X$ = A$: LET Z$ = "BOSQUE"
   GOSUB 9930: P = U
```

y agregue estas líneas:

```
9929 STOP
9930 LET U = 0: LET X = LEN(X$):
     LET V = LEN(Z$)
9940 FOR W = 1 TO L - V + 1: IF
     MID$(X$,W,V) = Z$ THEN LET U = W
9950 IF U <> 0 THEN LET W =
     = L - V + 1
9960 NEXT W: RETURN
```

En el Spectrum, sustituya MID$(X$,W,V), en la línea 9940, por FN M$(X$,W,V) y consulte el "Complementos al BASIC" anterior

■ **Bucles 2** ¿Qué se verá en pantalla si se ejecuta este programa?

```
10 REM
20 REM ESTE ES UN BUCLE DE TIEMPOS
30 REM COMPROBAR CUANTO DURA
40 REM
50 PRINT "START"
60 FOR X = 1 TO 5000
70 NEXT X
80 PRINT "STOP"
90 END
```

■ **Bucles 3** ¿Qué resultado se imprimirá si ejecuta este programa y, al requerírsele, digita el número 60?

```
10 PRINT "PIENSE UN NUMERO Y DIGITELO"
20 INPUT N
30 LET A = 100
40 FOR L = 1 TO N
50 LET A = A + 1
60 NEXT L
70 PRINT "AHORA EL VALOR DE A ES"; A
80 END
```

■ **Bucles 4** ¿Qué sucedería al ejecutar este programa?

```
10 PRINT "ME GUSTA EL BASIC"
20 GOTO 10
30 END
```

■ **Bucles 5** ¿Qué se vería en pantalla si se ejecutara este programa?

```
10 FOR Q = 1 TO 15
20 PRINT "ESTOY ALGO TONTO"
30 NEXT Q
40 END
```

■ **Read-Data 1** ¿Qué resultado se imprimiría?

```
10 READ X
20 READ Y
30 READ Z
40 PRINT "COMPARAMOS LA SENTENCIA 'READ'"
50 DATA 50,100,20
60 PRINT X + Y + Z
```

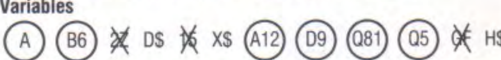
■ **Read-Data 2** ¿Qué se imprimiría en pantalla si se ejecutara este programa?

```
100 FOR L = 1 TO 10
110 READ X
120 PRINT "X = "; X
130 NEXT L
140 DATA 1,2,3,5,7,11,13,17,19,23
```

Las respuestas, en el próximo capítulo.

Respuestas a los "Ejercicios" de las páginas 136 y 137

Variables



Aritmética 1
10 LET B = 6
20 PRINT B

Aritmética 2
10 LET A = 5
20 LET B = 7
30 LET C = 9
40 LET D = A + B + C
50 PRINT D

Aritmética 3
17

Aritmética 4
25
25

Comparaciones 1
5

Comparaciones 2
601 (los enteros se suponen)

Comparaciones 3
10000

Print 1
PRINT "EL VALOR DE T ES";T

Print 2
640 PRINT "LO SIENTO, PERO SU PUNTUACION DE";S; "ES DEMASIADO BAJA"

Print 3
Se trataba de un error deliberado. El punto y coma al final de la línea producirá un error de sintaxis cuando se ejecute. El programa debe decir:

200 LET A$ = "¿MI COMPUTER?"
210 LET B$ = "¿LE GUSTA?"
220 PRINT B$;A$

Y entonces el resultado sería:
¿LE GUSTA MI COMPUTER?

Entrada 1
6

Entrada 2
POR FAVOR DIGITE SU NOMBRE
HOLA (SU NOMBRE) SOY SU ORDENADOR

Tenga en cuenta que las respuestas a las "variables" serán diferentes en algunas máquinas que no admiten más de un carácter alfabético (es decir, que no admiten sufijo numérico)

Jupiter Ace

El único ordenador personal de precio económico cuyo lenguaje de programación estándar es Forth en lugar de Basic; todo un desafío para programadores ambiciosos

El Jupiter Ace es una máquina para entusiastas de la informática y es uno de los pocos ordenadores cuyo lenguaje estándar no es el BASIC. Sus diseñadores habían formado parte del equipo del ZX Spectrum y decidieron producir un ordenador que a ellos mismos les agradara usar.

El lenguaje incorporado es el FORTH y ésta es la característica distintiva de esta máquina. Pero su hardware es lo suficientemente barato como para que cualquier persona interesada en aprender FORTH se pueda comprar el Ace en lugar de perfeccionar el ordenador que ya posea. El manual que viene con el ordenador constituye una excelente iniciación al lenguaje FORTH.

La línea del Ace se asemeja mucho a la de los modelos Sinclair. De hecho, su carcasa es del mismo plástico blanco y delgado que poseía el primer ordenador

Introducción al Forth

Versión en BASIC

```
100 REM UN PROGRAMA EN BASIC PARA IMPRIMIR
    'SHAZAM!'
110 FOR X = 1 TO 6
120 PRINT "SHAZAM!"
130 NEXT X
140 END
    RUN
```

Versión en FORTH

```
(UN PROGRAMA EN FORTH PARA IMPRIMIR 'SHAZAM!')
:SHOUT ." SHAZAM! " ; .
:CHORUS 6 0 DO SHOUT LOOP;
 CHORUS
```

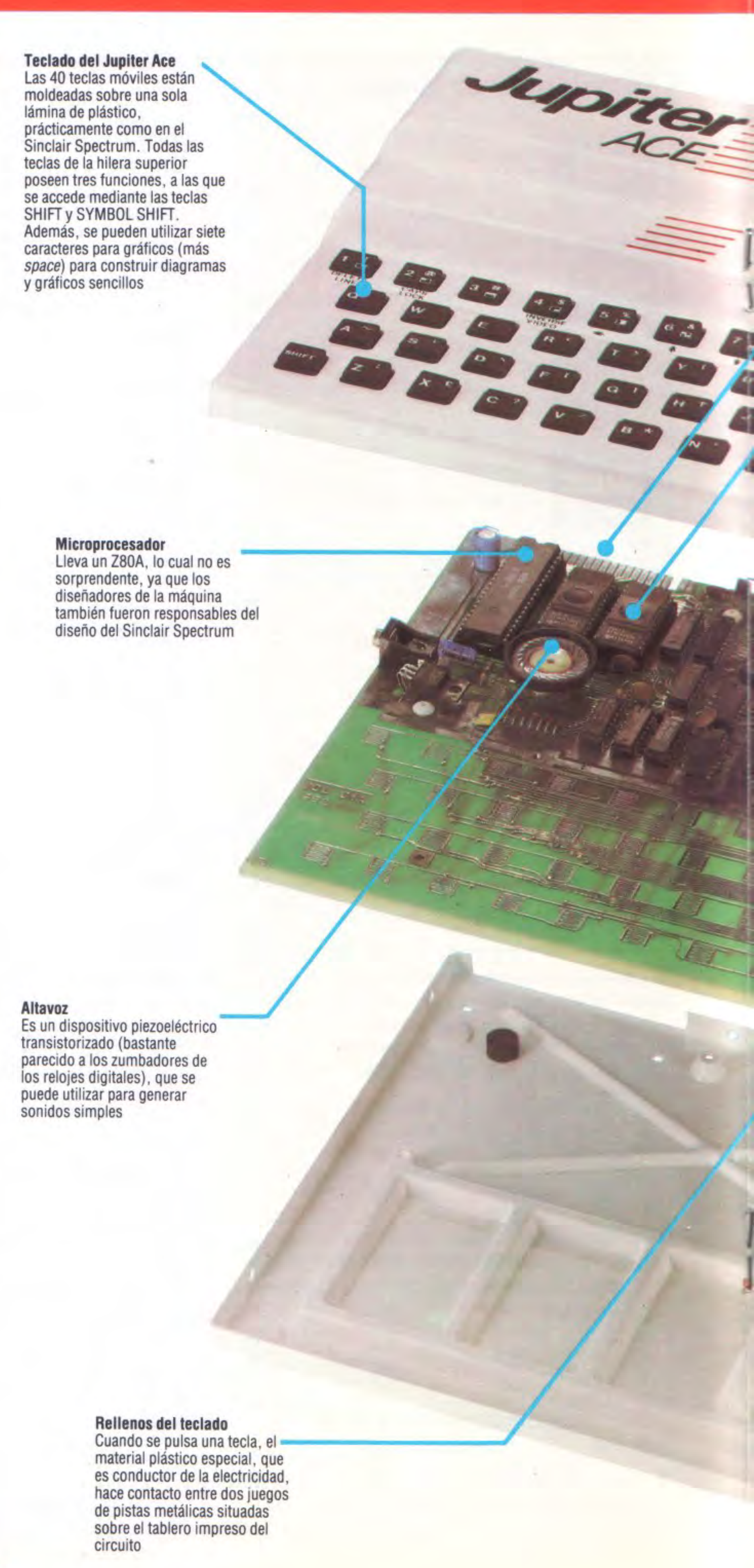
Estos dos programas harán exactamente lo mismo, pero la versión en BASIC semeja una receta, mientras el programa en FORTH ¡parece un sistema de escritura para adivinos!

El lenguaje FORTH comienza con un grupo de palabras órdenes (denominadas diccionario), y tiene la particularidad de que se le pueden agregar nuevas palabras. En nuestro programa en FORTH agregamos dos nuevas palabras al diccionario: SHOUT se define como una serie de caracteres a imprimir y CHORUS se define como una mezcla de "primitivas" (palabras predefinidas en el diccionario) y la nueva palabra SHOUT.

El FORTH también posee una memoria (denominada *stack*) y capacidad de procesar los números que ella contenga. El programa en FORTH efectúa las mismas operaciones aritméticas y lógicas que el programa en BASIC, pero no las lleva a cabo a través de expresiones algebraicas, sino manipulando el stack.

El lenguaje FORTH puede llegar a ser exasperante, algo así como un "cubo de Rubik"; no obstante, es innegable que constituye un eficaz método de programación y toda una nueva forma de pensar.

Lo que verdaderamente le confiere al FORTH todo su poder es su capacidad para definir y utilizar nuevas órdenes. En efecto, el usuario puede confeccionar el lenguaje de programación a la medida de la aplicación que está realizando. El FORTH es especialmente adecuado para programar dispositivos automáticos de uso doméstico, ya que el informático puede crear su propio diccionario de órdenes: MOVE (mover), FETCH (alcanzar), FIND (hallar), FOLLOW (seguir) y RETURN (retornar), por ejemplo



Teclado del Jupiter Ace
Las 40 teclas móviles están moldeadas sobre una sola lámina de plástico, prácticamente como en el Sinclair Spectrum. Todas las teclas de la hilera superior poseen tres funciones, a las que se accede mediante las teclas SHIFT y SYMBOL SHIFT. Además, se pueden utilizar siete caracteres para gráficos (más *space*) para construir diagramas y gráficos sencillos

Microprocesador
Lleva un Z80A, lo cual no es sorprendente, ya que los diseñadores de la máquina también fueron responsables del diseño del Sinclair Spectrum

Altavoz
Es un dispositivo piezoeléctrico transistorizado (bastante parecido a los zumbadores de los relojes digitales), que se puede utilizar para generar sonidos simples

Rellenos del teclado
Cuando se pulsa una tecla, el material plástico especial, que es conductor de la electricidad, hace contacto entre dos juegos de pistas metálicas situadas sobre el tablero impreso del circuito



Conexión para ampliación de memoria
Con un adaptador adecuado, el Ace también puede utilizar los paquetes de ampliación para el Sinclair ZX81

Lenguaje FORTH
Para retener el lenguaje se han empleado dos unidades EPROM de 4 Kbytes. El pedido de unidades ROM ha de efectuarse por millares de unidades, de modo que para volúmenes de fabricación más pequeños se utilizan unidades EPROM. La parte superior de cada EPROM posee una cubierta protectora; de lo contrario, cualquier luz ultravioleta borraría su contenido

Conexión dispositivos
Apta para una impresora u otra unidad

Reloj
Hace que el microprocesador funcione a 1 MHz

Modulador de RF
Proporciona una salida en blanco y negro sólo para televisor

Conexión cassette
Uno de los sistemas más fiables que se han desarrollado hasta ahora

Sistema de circuitos de video
Las máquinas que se venden en cantidades mayores, para realizar el mismo trabajo que este grupo de chips sólo necesitarían llevar un chip de diseño especial

Chris Stevens

Sinclair (el ZX80). Posee un pequeño zumbador transistorizado que puede producir una gama de tonos simples, pero con un mayor conocimiento del Ace se pueden crear sonidos más complejos. Al igual que todos los ordenadores Sinclair, el Ace posee una conexión para alimentación eléctrica, otra para televisión y dos grandes conexiones para cualquier otro elemento especial del equipo.

La cassette cuenta con una conexión sencilla de clavija bipolar, una de las interfaces para cassette más fiables que puede poseer un ordenador.

La visualización es en blanco y negro con 32 caracteres por línea, si bien su resolución para gráficos es de 64 por 48. El usuario puede volver a diseñar cualquiera de los caracteres para crear símbolos matemáticos específicos o formas determinadas para juegos.

La máquina goza ya de gran aceptación debido a que utiliza el lenguaje FORTH, y existen a la venta diversos dispositivos accesorios. Los programas en FORTH por lo general son muy cortos, de manera que en la memoria estándar, de 3 Kbytes, cabe una cantidad sorprendente de ellos. Para escribir programas más largos se necesita más memoria. Existen paquetes de RAM tanto de 16 Kbytes como de 48 Kbytes, aunque se pueden emplear los paquetes RAM para el ZX81 con un adaptador. Con éste se pueden usar con el Ace la mayoría de los accesorios Sinclair, siempre que se disponga de diversos tipos de entrada y salida, como puertas en paralelo y conexiones seriales.

JUPITER ACE

DIMENSIONES
215 × 190 × 30 mm

PESO
246 g

CPU
Z80A

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
RAM de 3 Kbytes ampliable exteriormente a 51 Kbytes; ROM de 8 Kbytes

VISUALIZACION EN VIDEO
Blanco y negro, líneas de texto de 32 × 22, gráficos de 64 × 48

INTERFACES
Conexiones televisión cassette y alimentación eléctrica (9 v); dos grandes conexiones: la primera tiene líneas completas de dirección y datos del procesador; la segunda posee líneas de datos y algunas de selección

LENGUAJE SUMINISTRADO
FORTH

OTROS LENGUAJES DISPONIBLES
Ninguno

VIENE CON
Unidad de alimentación eléctrica en el enchufe principal, cables para cassette y televisión

TECLADO
Relleno plástico similar al del Sinclair Spectrum pero más blando y menos preciso. Las teclas se han de pulsar exactamente por el centro. Todas las teclas disponen de repetición automática y dos teclas de cambio permiten producir todos los códigos de ASCII

DOCUMENTACION
Se trata, con toda seguridad, del mejor manual para ordenadores pequeños que se haya realizado, y haría enrojecer de vergüenza a la gran mayoría de los proveedores de ordenadores mayores. Su autor es el mismo que escribió los manuales del ZX81 y del Spectrum. En 180 páginas ofrece una introducción al FORTH y una descripción completa del Ace, y siempre con el apoyo de numerosos ejemplos. Dispone de una lista de contenidos, de cuatro apéndices que proporcionan rápidas referencias de todas las facilidades disponibles, y de un índice

Gráficos con vida

Las memorias de gran capacidad permiten producir con ordenadores personales imágenes plenas de color y movimiento

Una de las características más sorprendentes de los ordenadores personales es su capacidad para producir gráficos y visualizaciones en movimiento, comúnmente denominadas imágenes animadas. En la mayoría de los microordenadores, el usuario puede trazar puntos individuales, dibujar líneas y círculos y cambiar los colores del fondo y de los primeros planos.

Para los juegos de simulacro y de acción rápida, debemos ser capaces de simular el movimiento. La manera más sencilla de conseguirlo consiste en producir una serie de imágenes fijas, una después de la otra. Esto se debe realizar con la rapidez suficiente como para crear la ilusión de movimiento. Las imágenes de televisión se producen mediante un procedimiento similar al descrito.

Rapidez de acción

Otra forma de crear la ilusión de movimiento consiste en imprimir un carácter, borrarlo y volverlo a imprimir en una posición ligeramente alejada de la original. Para que el movimiento fluya uniformemente es imprescindible que la distancia entre un paso y otro sea mínima. Asimismo, el tiempo invertido en producir la forma y en borrarla debe ser lo más corto posible.

bres de parpadeos. Sumado a ello, se plantea el problema adicional de controlar los caracteres individuales a medida que aumenta su número en la pantalla.

Diversos ordenadores, en particular el Commodore 64, el Sord M5, el Texas Instruments TI99/4A y la gama Atari, superan este inconveniente y proporcionan una animación que utiliza las mismas técnicas que emplean las máquinas de juegos recreativos accionados con monedas. Esta técnica se conoce como *gráficos sprite*.

Los sprites son "objetos" o formas que pueden ser desplazados, independientemente unos de otros, a través de la pantalla. Esto se logra, sencillamente, cambiando el contenido de un par de direcciones de memoria, que especifican las coordenadas X e Y (las posiciones izquierda-derecha y arriba-abajo). Típicamente, X puede abarcar entre 0 y 255, e Y entre 0 y 191. Algunos sistemas permiten incluso especificar la velocidad y la dirección del movimiento de cada sprite, y el ordenador hace todo el resto.



Los sprites normalmente se realizan utilizando chips o



El tren de pensamientos

El tren que vemos en la ilustración se construyó a partir de tres sprites (la locomotora más dos vagones) utilizando en el Commodore 64 un paquete denominado Spritemaker. La imagen se creó a gran escala usando las facilidades de edición del paquete, y se guardó en una cassette, con las imágenes de la casa y del árbol. Una vez vueltos a cargar en la memoria del Commodore, los sprites se manipularon mediante órdenes POKE para determinar sus posiciones en la pantalla, su color y la velocidad del tren. También se especificó la "prioridad" de los sprites, de modo que el tren pasara por detrás de la casa pero por delante del árbol

Ian McKinnell

La utilización del BASIC para producir animación da como resultado un movimiento lento. Una forma de superar este inconveniente es recurrir al lenguaje ensamblador, método que requiere mucha práctica, atención y cuidado si desea obtener visualizaciones libres de parpadeos.

un sistema de circuitos de hardware especiales dentro del ordenador. Es posible comprar software para otros ordenadores con el fin de obtener resultados similares, pero por lo general esta solución resulta poco satisfactoria.

PLANO 0

PLANO 1

PLANO 2

PLANO 3

PLANO 4

Mark Watkinson

Los sprites se pueden mover a distintas velocidades, ya que cada uno de ellos está situado en lo que se conoce como un *plano de sprite*. Una escena de gráficos en pantalla se compone, por tanto, de una serie de planos apilados uno detrás del otro, si bien a simple vista uno los percibe en la pantalla como si se tratara de un solo plano.

El efecto tridimensional se obtiene haciendo pasar los sprites por delante o por detrás de los demás. Los sprites se numeran de 0 hasta el máximo número disponible que, por ejemplo, en el Sord M5 es 32. Cuando dos sprites se superponen se visualiza el que posee el número más bajo. De manera que si los sprites se ordenan cuidadosamente, se puede obtener con suma facilidad un efecto tridimensional: por ejemplo, si se hace pasar un tren por delante de un árbol, éste se oscurecerá a medida que el tren vaya avanzando. Por otra parte, el mismo tren se puede neutralizar parcialmente si se lo hace pasar por detrás de una casa que posea un número más bajo.

Utilizando la gama de colores que ofrezca el ordenador, cada sprite se puede colorear de forma individual. A veces se puede ampliar o reducir un sprite si se cambia el contenido de una dirección de memoria.

Con todo, es en los programas de juegos donde los sprites demuestran toda su valía; la *detección de colisiones* es una configuración particularmente útil. Cuando se superponen dos o más sprites (por ejemplo, cuando su misil entra en contacto con la nave espacial enemiga), el sistema puede ser programado para que salte hasta otra parte del programa con el fin de crear los gráficos para una explosión.



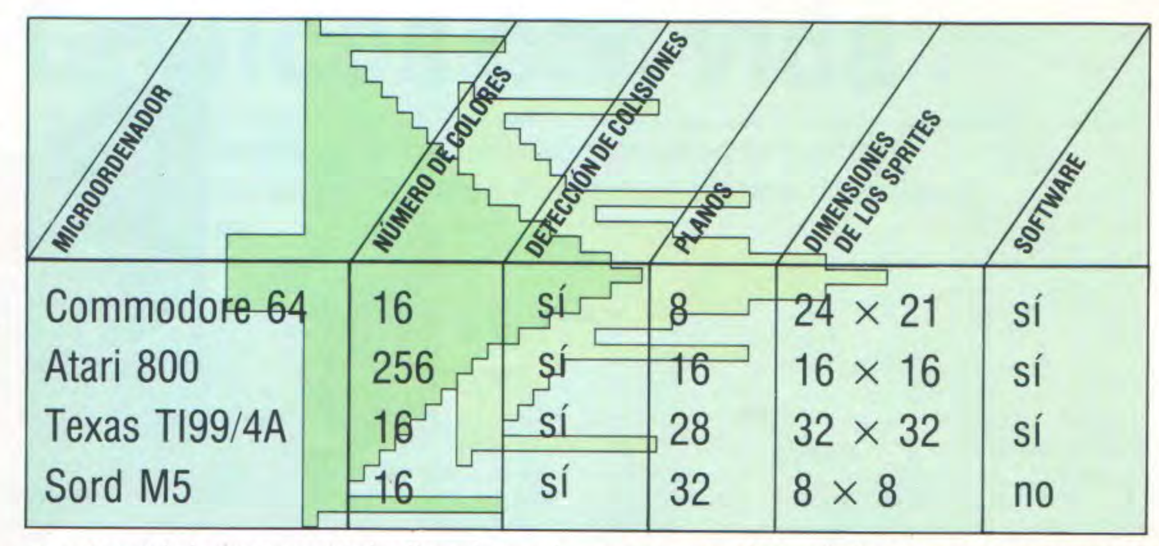
Antes de utilizar los sprites es preciso crearlos, y el modo de hacerlo se asemeja mucho a la forma en que se diseña un nuevo carácter. Las letras del alfabeto, los números y los símbolos especiales para gráficos están almacenados dentro del ordenador, en un chip que se conoce como *generador de caracteres*.

Los caracteres, como ya hemos mencionado en otro apartado de esta obra, por lo general se construyen en una matriz de puntos (o *pixels*) de ocho por ocho. Las dimensiones máximas de los sprites varían de una máquina a otra, pero generalmente su ancho y su alto corresponden a los de varios caracteres. En el Commodore 64, por ejemplo, la dimensión máxima es 24 pixels de ancho por 21 de alto.

La mejor forma de realizar un sprite consiste en tra-

En diferentes planos
Utilizando los gráficos sprite, se puede crear una compleja imagen al situar los elementos componentes en distintos planos, uno detrás de otro. La inmensa ventaja que ofrece este sistema es que los objetos de los diferentes planos se pueden mover de forma absolutamente independiente respecto a los demás, aunque en la pantalla se perciban como si se tratara de un solo plano. El programador les asigna a los planos un orden de prioridad, de modo que cuando en la pantalla se superponen dos sprites, el que posee la prioridad más alta es el que se ve por delante del otro, obteniéndose así un efecto tridimensional
La mayoría de los ordenadores que disponen de gráficos sprite también pueden disponer que el programa se interrumpa cada vez que entran en contacto dos sprites, para crear la consiguiente explosión o aumentar el puntaje del jugador

Dimensiones del sprite
Sólo cuatro de los microordenadores personales más populares disponen de gráficos sprite: el Commodore 64, el Sord M5, el TI99/4A y la gama Atari.
En el caso del Atari, los 16 colores que normalmente se ofrecen pasan a ser 256, ya que cada color posee 16 grados de "luminosidad".
Cuando se utilizan para juegos, a menudo es interesante saber cuándo dos sprites se tocan (por ejemplo, al chocar dos naves espaciales), y por ello se proporciona una configuración para "detección de colisiones". El efecto tridimensional se crea colocando cada sprite en un plano diferente; lo mejor para un sistema es que disponga de la mayor cantidad de planos posible. Las dimensiones máximas de cada sprite se expresan en pixels. Los sprites se pueden ampliar o reducir, y desplazar por la pantalla.
Para simplificar el proceso de construcción de los sprites existen paquetes especiales de "programas de utilidad"



Microordenador	Número de colores	Detección de colisiones	Planos	Dimensiones de los sprites	Software
Commodore 64	16	sí	8	24 × 21	sí
Atari 800	256	sí	16	16 × 16	sí
Texas TI99/4A	16	sí	28	32 × 32	sí
Sord M5	16	sí	32	8 × 8	no

zar una cuadrícula que represente las dimensiones máximas, y construir la forma deseada rellenando los cuadrados adecuados. Ya se sabe que los ordenadores trabajan utilizando solamente unos y ceros (véase p. 28). Los cuadrados en color se deben representar como unos y los cuadrados en blanco como ceros.



El ordenador maneja la mayor parte de la información en forma de bytes (grupos de ocho bits). En el manual del ordenador se le explicará cómo se debe separar por grupos de ocho la cuadrícula entera de puntos que compone un sprite. Cada byte, a su vez, ha de convertirse en un número decimal, entre 0 y 255 como máximo, antes de que se lo pueda utilizar para un programa en BASIC. Esto se consigue multiplicando el dígito binario (bit) situado más a la izquierda por 128, el siguiente por 64, y así sucesivamente. Luego se suman los resultados.

La serie de números decimales resultante define por completo el diseño del sprite. Estos números se colocan en direcciones de memoria empleando un programa en BASIC; el procedimiento exacto a seguir varía de una máquina a otra. Luego el ordenador requiere que se le proporcionen instrucciones acerca de en qué lugar de la memoria puede hallar las especificaciones para cada uno de los sprites requeridos.

Entonces se puede realizar lo que se desee utilizando órdenes simples para especificar la posición normal de cada sprite en la pantalla, para cambiar su color, agrandarlo o reducirlo y detectar cuándo se superponen dos sprites o más.

Existen paquetes de software, denominados *programas de utilidad*, para la mayoría de las máquinas con sprites. Estos programas logran que el proceso de creación de la imagen resulte menos tedioso. Visualizan la cuadrícula en pantalla a gran escala y permiten dibujar la imagen simplemente moviendo el curso intermitente a través de la cuadrícula. Toda la aritmética se efectúa de forma automática y luego los resultados se colocan en los bytes adecuados. Por último, la cuadrícula desaparece y aparece en pantalla el propio sprite, listo para ser manipulado.

Los sprites han revolucionado los gráficos de los ordenadores personales al proporcionar un método sencillo y eficaz de producir visualizaciones con movimiento rápido y a todo color.



Ha nacido un sprite

Para construir un sprite, lo mejor es comenzar con un trozo de papel cuadriculado. El objeto se dibuja en él rellenando algunos de los cuadrados. Algunos ordenadores pueden realizar sprites multicolores, si bien aquí, por razones de simplicidad, hemos ilustrado un sprite de un solo color. En una segunda cuadrícula, los cuadrados en color se representan como unos y los blancos como ceros. Y como la memoria del ordenador está dividida en bytes (de ocho bits cada uno), toda la cuadrícula se ha de dividir en grupos de ocho cuadrados.
Cada grupo se ha de convertir en un número decimal. El bit situado más a la izquierda se multiplica por 128, el siguiente por 64, y así sucesivamente. Estos resultados se suman luego para obtener un resultado que abarca entre 0 y 255. Éste se utiliza después en el programa en BASIC

Mark Watkinson

Steve Wozniak



Apple Computer Inc.

De bromista autodidacta de la electrónica a multimillonario: cómo se crearon los populares ordenadores Apple

En Estados Unidos, el gran público conoce más a Steve Wozniak por la puesta en escena de multitudinarios conciertos de rock al aire libre que por ser el hombre que construyó, él solo y en el garaje de su casa, los Apple I y II. Pero en el mundo de la informática se le considera el "genio de la electrónica" que más ha hecho por simplificar y popularizar el microordenador. Las máquinas de Wozniak fueron las primeras en disponer de color, gráficos, teclado y video como configuraciones estándar. Y el Apple II es tan popular que ya se han vendido un millón de unidades.

La meteórica ascensión de Steve Wozniak, desde ingeniero en el garaje de su casa hasta multimillonario en dólares, parece un moderno cuento de hadas. Wozniak nació y creció en el ahora famoso lugar de nacimiento del chip, Silicon Valley (California). Su padre, ingeniero de profesión, le enseñó la ley más elemental de la electrónica, la ley de Ohm, pero a partir de ahí Wozniak aprendió electrónica por sí mismo.

De niño ocupaba su tiempo jugando con componentes electrónicos y aplicando su genio técnico en hacer travesuras en la escuela. Una vez construyó un dispositivo electrónico llamado caja azul, idea que concibió inspirándose en el personaje de una revista que era un proscrito de la tecnología. El dispositivo podía imitar ciertos tonos a través del sistema telefónico. Estos tonos especiales eran la indicación de que la persona que telefoneaba había insertado en la caja de monedas del teléfono la cantidad de dinero necesaria. Esto le permitió a Wozniak hacer llamadas telefónicas gratis a todo el mundo. Habló con amigos de Inglaterra, y por este procedimiento llamó incluso ¡al Papa!

Wozniak jamás estudió la carrera de ingeniería; en la escuela era un alumno excelente en matemáticas y en electrónica, pero abandonó los estudios. Su primer trabajo fue como técnico en la inmensa corporación norteamericana Hewlett Packard, donde diseñó calculadoras. Pero ellos le dijeron que no estaba capacitado para hacer lo que él realmente deseaba: diseñar ordenadores. De modo que empezó a trabajar por su cuenta, especialmente por la noche, y diseñó un microordenador que la Hewlett Packard rechazó. Wozniak no se desalentó; dejó la compañía, y con Steve Jobs, condiscípulo suyo en la escuela (y compañero de travesuras), construyó y vendió 50 unidades de la máquina que había diseñado. Así nació el Apple I. Bautizaron con el nombre de Apple ("manzana") al ordenador y a la empresa simplemente porque Jobs una vez había trabajado en una huerta.



Durante 1975 y 1976 Wozniak se encerró en su garaje. Allí las ráfagas de inspiración de su genio lo impulsaban a trabajar de noche y de día, hasta que finalmente produjo el Apple II. Tenía 26 años. Aún hoy los expertos consideran la creación del Apple como una proeza sorprendente, por su diseño y sistema de circuitos tan brillantemente sencillos.

Afirman que Steve Wozniak lee el sistema de los diagramas de circuito y sincronización de un chip con la misma facilidad con que algunas personas leen el futuro en las hojas de té. Una de las innovaciones importantes del Apple II fue la simplificación de la unidad de disco. Antes de Wozniak ésta requería 30 chips, pero él la rediseñó para que sólo incluyera cinco en el caso del ordenador personal. No se trata tanto de que Wozniak hubiera creado algo completamente nuevo, sino que simplificó y comprimió tanto todos los componentes, que consiguió que pronto cualquiera pudiera usar un ordenador en su propia casa.

Wozniak nunca tuvo la intención de dedicarse a los negocios. Fue su socio Steve Jobs quien se encargó de vender los Apple y de crear la Apple Corporation. En la actualidad la empresa cuenta con 3 300 empleados en todo el mundo y se dice que gracias a ella al menos 50 personas se han hecho millonarias. Wozniak sólo posee el 4 % de las acciones de la Apple y jamás ha participado (ni le ha interesado hacerlo) en la gestión empresarial. Él sigue prefiriendo jugar con los ordenadores y soñar con ideas nuevas.

En el verano de 1983, después de un paréntesis de dos años durante los cuales se dedicó principalmente a organizar festivales de rock, Wozniak se reintegró en la Apple para trabajar en algunos proyectos nuevos. Nadie sabe con certeza en qué está trabajando. Al parecer está escribiendo algunos programas de aplicaciones. También está dirigiendo un proyecto del Apple II que aspira a crear un sistema personal de producción de video y edición, con gráficos de alta calidad, capaz de crear dibujos animados. Pero a Wozniak no sólo le interesa conseguir que las máquinas ya existentes sean mejores y más rápidas. También piensa que en el futuro se podrá diseñar un microordenador muy inteligente. Empleando solamente un sencillo programa este microordenador sería capaz de aprender cualquier materia.

El presidente
Steve Jobs, presidente de la Apple Computer Inc., era condiscípulo de Steve Wozniak y también su amigo y compañero de travesuras. Si Wozniak era el genio electrónico del equipo, Jobs era el niño precoz de los negocios. En 1975, cuando se diseñó el Apple I, Jobs tenía 20 años de edad. Él adivinó el potencial que ofrecía la venta de la máquina, entre las personas aficionadas a la informática, pero vio muy pronto que la podría introducir en un nuevo mercado, el de los ordenadores personales. Wozniak fue el creador del Apple, pero fue Jobs quien fabricó y vendió el aparato a nivel masivo. El Apple II se embaló en una caja, completo y listo para conectarlo en una casa particular, donde cualquiera pudiera usarlo. Mientras Wozniak y Jobs aún trabajaban en el taller del garaje, se invirtió en la empresa el primer capital: 1 350 dólares, provenientes de la venta de la furgoneta Volkswagen de Jobs y de la calculadora programable de Wozniak. Sin embargo, enseguida un joven millonario, Mike Markkula, prestó su apoyo financiero a lo que él consideraba un proyecto interesante.
Cuando la existencia de la empresa se hizo pública, en 1980, fue tema de candente actualidad. De inmediato la compañía demostró su potencial al vender ordenadores por valor de 583 millones de dólares sólo en 1982.
El fabuloso crecimiento de la Apple Corporation ha sido ampliamente reconocido y recompensado en Estados Unidos. La revista *Time* le dedicó su portada a Steve Jobs, reconociéndolo como uno de los hombres de negocios más joven y de mayor éxito de Norteamérica

Varillas mágicas

Un lápiz óptico es un accesorio que sirve para dibujar o marcar en la pantalla. Vamos a explicar cómo funcionan estos extraordinarios dispositivos

La "aversión a los teclados" es una de las causas más comunes por las que la gente se muestra reacia a familiarizarse con los ordenadores, ya sea en su casa o en la oficina. Debido a que el teclado se asemeja al de una máquina de escribir y a que estas personas no saben mecanografía y, más aún, comprueban que el teclado contiene diversas teclas con signos desconocidos, temen hacer el ridículo. La solución a este problema es el lápiz óptico (junto a otros dispositivos tales como entrada de voz), que posee asimismo otras funciones.

Un lápiz óptico es un dispositivo cilíndrico (su nombre deriva de su gran parecido con un lapicero corriente) de uno de cuyos extremos sale un cable en espiral similar al de los teléfonos. En el otro extremo de este cable hay un enchufe que se conecta en la parte posterior del ordenador. Cuando se apunta directamente a la pantalla con el lápiz óptico (en algunos ordenadores el lápiz ha de tocar ésta para activar un interruptor situado en su interior), el ordenador detecta la posición exacta que está señalando el lápiz.

Lo que sucede en realidad es que un fotodetector localizado en la punta del lápiz responde con un impulso eléctrico cuando pasa por el punto de luz que continuamente explora toda la pantalla para crear la imagen. El sistema de circuitos en el interior del chip controlador del video calcula dónde estaba el punto de exploración en el momento en que el lápiz óptico emitiera su señal.

El lápiz óptico, por lo tanto, se utiliza básicamente para seleccionar un ítem visualizado en pantalla. Sabiendo qué punto está señalando el lápiz, el ordenador deduce el carácter o el símbolo del que forma parte ese punto. Muchos programas de aplicaciones incorporan "menús" en su funcionamiento. Un menú es simplemente una lista de opciones que se visualiza en la pantalla, entre las cuales el usuario debe hacer su elección, como haría con el menú de un restaurante. En un programa de economía doméstica, el menú sería:

1) Efectuar un pago
2) Examinar el estado de cuentas bancario
3) Aceptar un recibo
etcétera.

Normalmente el usuario debería indicar la acción que ha escogido pulsando una tecla (1, 2 o 3) o digitando una palabra de orden. Con un lápiz óptico, todo cuanto ha de hacer será señalar la opción requerida. El ordenador por lo general responde haciendo centellear esa opción determinada, para indicar que ha aceptado la entrada. Algunos programas más sofisticados funcionan casi completamente mediante este tipo de menús (se los suele describir como programas de opciones de trabajo), en cuyo caso el usuario sólo toca el teclado cuando se requiere una información verdadera, como por ejemplo el nombre y la dirección de alguien.



Lente
La cantidad de luz que emite un pixel al ser refrescado es tan pequeña, que se debe utilizar una lente para concentrarla en la superficie del fotodetector

Rutinas especiales

La dificultad que suponen los programas de este tipo es que se deben escribir pensándolos especialmente para que trabajen con el lápiz óptico y no desde el teclado. En realidad la cuestión estriba en escribir una pequeña rutina que asuma las coordenadas que indican la posición normal del lápiz óptico desde el controlador de video y que determine cuál de las opciones se produce en la pantalla en esa posición. Lamentablemente, son pocos los fabricantes de software que proporcionan versiones de sus programas que puedan ejecutarse con lápiz óptico.

Sin embargo, además de utilizarse para seleccionar ítems, los lápices ópticos se pueden emplear para crear imágenes en pantalla. La mayoría de los ordenadores personales para los cuales existen lápices ópticos poseen un paquete de programas apto para tal fin. Al usuario se le presenta una pantalla en blanco (sobre la cual él puede dibujar un diagrama, un dibujo o un esbozo) y una sección separada (normalmente a lo largo de la parte inferior de la pantalla) que le proporciona una serie de funciones especiales para ayudarlo en el proceso de creación. Una de éstas puede ser una paleta de colores, que funciona igual que una paleta de óleos como la que emplean los pintores. El lápiz óptico se coloca sobre el próximo color requerido, y el lugar por donde después se pase el lápiz por la pantalla principal dejará una línea de ese color.

De la parte inferior de la pantalla, utilizada como paleta, el usuario también puede seleccionar distintas





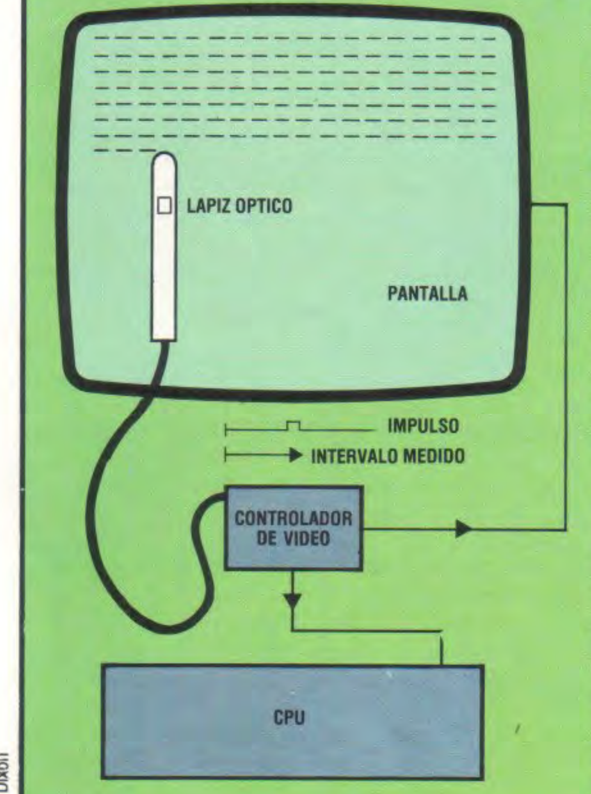
Kai Choi

Fotodetector
Se trata de un dispositivo semiconductor que, por decirlo de alguna manera, es como un transistor o un diodo al que se le hubiera aserrado la punta. La luz que recibe este dispositivo controla la corriente eléctrica que fluye a través del mismo

Circuitos de amplificación
Éstos sirven para detectar y amplificar la corriente que pasa a través del detector y para enviarle una señal de retorno adecuada al chip controlador de video del ordenador. Algunas veces estos circuitos están alojados fuera del lápiz propiamente dicho

Interruptor
La mayoría de los lápices ópticos incorporan alguna clase de interruptor, a veces para operar manualmente, pulsándolo con un dedo, y otras veces para activar presionando el lápiz óptico contra la pantalla. El interruptor es necesario para que el lápiz no reaccione a la luz (por ejemplo, a la luz de la habitación) cuando no se lo está utilizando para seleccionar un ítem en la pantalla

Cable
Este cable en espiral al estilo de los del teléfono se enchufa directamente en la parte posterior del ordenador y desde allí conecta con el chip controlador de video



Cómo funciona
El funcionamiento del lápiz óptico suele depender del chip controlador de video situado en el interior del ordenador. Este chip, explorando la pantalla muchas veces por segundo con líneas horizontales, produce imágenes de video en el televisor o en el monitor (esta acción se denomina *raster scan*). Los televisores españoles tienen 625 líneas en la pantalla y la frecuencia de exploración es de 50 por segundo. La salida del lápiz óptico también se deriva directamente a este chip. Cuando se señala un ítem de la pantalla con el lápiz óptico y se activa el interruptor, el controlador de video se limita a medir el tiempo que media desde la última vez que comenzó a trazar las líneas en la parte superior de la pantalla y el instante en que recibió el impulso proveniente del lápiz óptico. A partir de esta información calcula sobre qué línea está situado el lápiz y a qué distancia se halla esta línea. Esta información se convierte luego en las coordenadas X e Y que necesita el programa

Liz Dixon

anchuras de trazo y texturas y, asimismo, puede dibujar círculos y cuadrados. En resumen, todo cuanto exponíamos en "El artista electrónico" (véase p. 26) se puede realizar con un lápiz óptico y, probablemente, mucho más rápido que mediante el teclado.

Asimismo, están comenzando a aparecer en el mercado juegos que emplean lápices ópticos. Capturar monstruos extraterrestres es mucho más fácil con un lápiz óptico que con el teclado, de manera que los juegos han de incorporar otros grados de dificultad. Otros juegos más sosegados, como el ajedrez informatizado, también se pueden beneficiar de la utilización del lápiz óptico: uno señala adónde quiere desplazar el caballo y el ordenador se ocupa de todo lo demás.

Quizá el grupo más numeroso de usuarios de lápices ópticos sea, no obstante, el de ingenieros y diseñadores. Los sistemas de CAD (Computer Aided Design: diseño auxiliado por ordenador) se emplean bastante en los anuncios publicitarios de nuevos modelos de coches; en realidad, son sistemas informatizados como tantos otros, pero con un software especializado y gráficos de gran calidad. Si se utiliza un sistema CAD para diseñar un nuevo dispositivo electrónico, la pantalla incorporará la representación de todos los componentes a los que el diseñador pueda desear acceder; éste puede "escoger" cualquiera de ellos y colocarlo en el lugar de la pantalla que crea conveniente.

El lápiz óptico constituye uno de los mejores ejemplos de accesorio para ordenador que, además de poseer gran valor práctico, resulta fácil y divertido de emplear.

Planificando el futuro

	A	B	N
1	ENERO	FEBRERO	TOTALES
2	42,41	18,75	388,4
3	160,35	149,89	1732,7
4			

Campos de juego
Una hoja electrónica está dividida en líneas y columnas; la intersección de una línea con una columna se denomina *campo* o *celda* y se puede direccionar a partir de sus coordenadas (A1, B3, etc.). Cada campo puede contener un título (p. ej., ENERO), un número (p. ej., 149,89) o una fórmula. El campo N2 contiene la fórmula 'SUM(A2: M2)', que es la suma de las cifras de la línea superior, de enero a diciembre. El resultado de este cálculo está visualizado: 388,4. Observe que se ha dividido la hoja electrónica en dos ventanas, de manera que las cifras de marzo-diciembre no aparecen visibles

Un programa de "hoja electrónica" puede ayudar a darle a su ordenador una utilidad lucrativa. Sus principales aplicaciones son planificación, presupuesto y previsión financiera

Se ha calculado que los gerentes dedican aproximadamente el 30 % de su tiempo a la elaboración de presupuestos, una actividad que siempre implica considerar las respuestas a muchas preguntas del tipo "¿qué ocurriría si...?" Tradicionalmente se utilizaba para ello una hoja de papel, por lo general equivalente al tamaño de una doble página de *Mi Computer*. En ella se trazaban verticalmente una docena o más de columnas, cada una de las cuales se encabezaba con el número de un mes, y todos los tipos de gastos uno debajo del otro sobre un costado. En cada columna se incluían los gastos mensuales en las diversas categorías. Siguiendo este método, uno podía sumar cada columna para obtener los gastos totales mes por mes, y sumar individualmente todas las líneas para hallar el total gastado durante un año entero en cada rubro.

El problema se planteaba cuando uno había planificado gastar demasiado (o, peor aún, muy poco) y debía volver atrás y alterar una gran cantidad de cifras, y volver a calcular en función de ellas los totales de las líneas y las columnas.

Utilizando un programa de hoja electrónica, uno puede volver a calcular la página entera de cifras cada vez que se altera un único elemento básico. Si, por ejemplo, ha variado el costo del transporte del mes de enero, esta variación alterará los gastos totales de ese mes; así como el total de los gastos correspondientes a ese rubro, con sólo tocar una tecla. ¡No hay que sorprenderse de que los paquetes de hojas electrónicas sean el tipo de programas que más se venden en todo el mundo!

Al igual que la mayoría de los programas comerciales, las hojas electrónicas por lo general se escriben con *overlays*, es decir, que en realidad no todo el programa reside realmente en la máquina todo el tiempo. Si se considera que el programa está dividido en subrutinas (véase p. 77), la subrutina que no se requiera en la operación normal no será recuperada del almacenamiento de apoyo (disco o cinta) hasta que se la "llame". El sistema operativo *colocará* entonces esa subrutina *sobre* sobre la que se ha vuelto redundante (de ahí el nombre *overlay*: sobreponer). Como se puede imaginar, este método para ampliar la memoria disponible es verdaderamente muy útil, pero implica que a menudo uno debe esperar mientras la información se transfiere desde el medio de almacenamiento de apoyo a la memoria principal.

Los paquetes de hojas electrónicas (por lo general se los puede reconocer por su nombre, ya que la mayoría de las veces terminan con la partícula *calc*) existen para una gran variedad de ordenadores personales y de gestión empresarial. El más popular en Gran Bretaña, en función de su volumen de ventas, es el Visicalc, escrito originalmente para el Apple II y que salió a la venta a mediados de 1979. El mercado de software para microordenadores reacciona con mucha rapidez ante el éxito obtenido por uno de sus productos, y este caso no fue una excepción. Antes de que uno pronunciara las palabras "hoja electrónica", ya se hallaban a la venta, a lo largo de todo el país, programas para cualquier clase de máquina, apropiados o no.

Para que una hoja electrónica sea realmente útil ha de satisfacer dos requisitos: dimensiones (no necesariamente las que se ven en la pantalla, porque, como veremos más adelante, lo que se contempla es sólo una "ventana" de la totalidad) y una buena gama de órdenes para control y formato.

Esto significa que existen severas limitaciones en cuanto a las clases de máquinas que tienen posibilidades de ejecutar un programa de hoja electrónica aprovechándolo al máximo. Por regla general, las exigencias mínimas son 32 Kbytes de RAM y una pantalla de 80 caracteres en el caso de una aplicación de gestión, si bien una pantalla de 40 caracteres probablemente será suficiente si se trata de fines domésticos.

Los usuarios de ordenadores personales comprobarán que muchos de los paquetes disponibles para máquinas basadas en cassette, como el Spectrum y el Vic-20, si bien por su propia naturaleza son de dimensiones y potencial limitados, también resultarán muy útiles.

Dado que las hojas electrónicas poseen la capacidad de responder a preguntas del tipo "¿qué ocurriría si...?", es obvio que se pueden emplear para elaborar modelos simples informatizados (véase p. 101). Allí dábamos un ejemplo del trabajo del analista de sistemas; si se diera el caso de que usted utilizara un paquete de hoja electrónica, comprendería enseguida la evidente necesidad de efectuar una planificación similar e igualmente cuidadosa. Hemos comentado que las bases de datos consisten en una cantidad de información no clasificada que se ordena de acuerdo a los requerimientos del usuario cuando se recupera la información. Existen procesadores de textos (otro software de gran demanda) que le permiten al usuario desplazar palabras o bloques enteros de texto en el momento de ejecución. Pero las hojas electrónicas son un poco diferentes, en el sentido de que en realidad exigen por parte del usuario un proceso de planificación.

La línea inferior
El cursor es el bloque rectangular que normalmente ocupa el campo N2. Si se digita algo, lo digitado aparecerá en el campo donde estaba situado el cursor. El contenido completo de ese campo también se visualizará en la línea de órdenes de la hoja electrónica, que en este caso está en la parte inferior de la página

FORMULA: SUM (A2:M2)

Por ejemplo, si está utilizando una hoja electrónica para analizar sus gastos, quizá le interese agrupar todos los gastos relacionados con la vivienda (pagos de alquiler o de hipoteca, impuestos, seguro, etc.) y luego emplear el resultado de ese cálculo en una tabla mayor dentro de la misma hoja. Habrá de ser muy cuidadoso al sumar todos los gastos relacionados con la casa antes de trasladar ninguna cifra a la tabla mayor.

Cada dirección individual de la hoja electrónica, denominada celda, se direcciona y localiza en función de sus coordenadas X e Y. Horizontalmente se utilizan las letras A - Z, AA - AZ, e incluso BA - BM, para cubrir la anchura total posible de la hoja, que en las versiones más corrientes es de 65 celdas. Verticalmente se pueden emplear los números del 0 al 255. Cada celda puede contener una etiqueta (por ejemplo "ventas" o "ganancia"), un valor al que se le da entrada como resultado de un cálculo, o bien derivado del mismo (como 1 000), o la fórmula para efectuar ese cálculo, como B4 + B6 * B5. Las fórmulas, debido a que suelen exceder las dimensiones de visualización del recuadro, normalmente se visualizan en una línea separada en la parte superior de la pantalla. Al comenzar una hoja electrónica nueva las dimensiones de la celda estarán preestablecidas, quizás en ocho o nueve dígitos o caracteres. Esto se conoce como dimensiones por omisión. Por lo general, es posible acortar o alargar las celdas para adaptarlas al tipo de cálculo que se esté realizando. Algunos paquetes permiten que la columna situada más hacia la izquierda (normalmente sus títulos o descripciones) sea más ancha que las demás. Y no es necesario decidir de inmediato el tamaño que se desea darles a las celdas. La mayoría de las versiones son susceptibles de ampliar o reducir, incluso después de haber colocado información en ellas. En caso de que se redujeran las dimensiones por debajo de la longitud del contenido, la parte no visualizada no se borra sino que solamente queda oculta a la vista.

El último de los componentes principales de la hoja es la línea de orden, que aparece en la parte superior o inferior de la pantalla en respuesta a la tecla de "orden": /, por ejemplo. Estas órdenes se emplean para dar su formato y manipular el trazado de la hoja en sí misma, no de la información, si bien puede llegar a incidir en la forma en que se presenta ésta. La mayor parte de las versiones más populares de programas para hojas electrónicas permiten efectuar una gran variedad de operaciones con la base de datos. Se puede, por ejemplo, borrar, trasladar o copiar líneas o columnas enteras, y además dividir la ventana para que se visualicen juntas partes de la hoja que normalmente están muy alejadas entre sí como para aparecer juntas en la ventana; después es posible también hacer mover en espiral (mover a través) estas ventanas por separado.

El movimiento entre las celdas se suele realizar mediante la tecla de control del cursor; no obstante, otra tecla de orden permite saltar hasta otra celda determinada. Cargar y guardar, borrar y proteger, todo ello se efectúa empleando las teclas de órdenes, y en este punto vale la pena volver a recalcar la importancia de guardar regularmente el trabajo que uno haya realizado. Componer una hoja electrónica es una tarea que toma mucho tiempo, por lo regular mucho más que el que se necesita para dar entrada a los datos. Por regla general, *siempre* se ha de guardar una hoja antes de empezar a dar entrada a los datos. Luego, en el caso de que se incurriera en un error garrafal, el haberla conservado le evitará una pérdida irreparable.

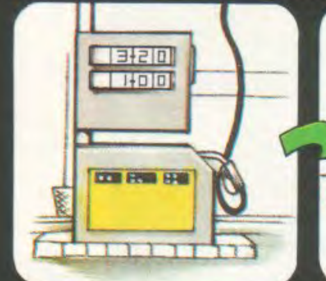
Los resultados se transmiten a la impresora empleando una tecla de orden, pero el usuario ha de asegurarse de definir la parte de la hoja que desea imprimir mediante la utilización de los parámetros. Así como la pantalla es una ventana, o un fragmento de la totalidad de la hoja, la misma función desempeñará la hoja impresa a través de una impresora de salida. Si desea imprimir una hoja más ancha que su impresora, lo aconsejable es hacer dos impresiones y luego unir las dos páginas.

Como ya hemos apuntado, el empleo de ventanas permite que se visualicen simultáneamente dos partes diferentes de la hoja. También se pueden imprimir las hojas con ventana dividida. Esto es especialmente útil al dar entrada a la información, puesto que uno puede hacer una referencia a una entrada anterior. La mayoría de los paquetes le permiten al usuario "retener" la línea superior o la primera línea, o ambas a la vez, que él crea por algún motivo, tomando en cuenta que por lo general éstas contienen los títulos o etiquetas.

Hasta este momento hemos venido considerando la hoja como una tabla a la cual uno sólo puede acceder en serie (un ítem después del otro, a lo largo de una línea o de una columna); pero si uno no desea seguir el trazado normal, sumando líneas o columnas, no existe ningún impedimento para trazar la hoja de cualquier otra forma si procediendo de esa manera se facilita la

Kevin Jones

¿Qué ocurriría si...?
Si se modifica el contenido de cualquier campo, la hoja electrónica volverá a calcular automáticamente todos los campos que dependan de esa cifra. La velocidad y sencillez de este proceso estimula al usuario a verificar la validez de sus proyectos, para comprobar lo que sucedería con los beneficios globales si variaran determinadas condiciones. Tomemos el ejemplo de un vendedor de frutas...



Si el precio de la gasolina aumentara en un X %...

Luego los costes mensuales de transporte se incrementarían en un Y %...


Esto, a su vez, haría subir el precio al por mayor de las manzanas...



Y ello incidiría directamente en el consumidor a través del aumento de los precios...

Mark Watkinson

fluidez de pensamiento del usuario. No obstante, conviene recordar que este tipo de sofisticación dentro de la base de datos requerirá un análisis de los sistemas mucho más exhaustivo que el que normalmente se lleva a cabo.

Las hojas electrónicas para gestión empresarial, como Visicalc, Supercalc y Masterplan, le ofrecen al usuario la posibilidad de pasar información desde la hoja electrónica a paquetes de gestión para bases de datos o tratamiento de textos, y existen muchos programas complementarios para que las salidas se realicen en una variedad de formas gráficas: circulares, por ejemplo, o de barras.

Ya hemos mencionado una de las posibles aplicaciones de las hojas electrónicas para el usuario de ordenadores personales: el análisis de los gastos y el presupuesto del hogar. Otra aplicación muy adecuada para el hogar sería en la instalación de la calefacción central, para la cual se han de considerar gran cantidad de variables: el tipo de combustible que se empleará, la cantidad y la clase de radiadores, la salida de la caldera, etc. De hecho, utilizar una hoja electrónica es de gran ayuda para cualquier proceso que implique tomar una serie de decisiones, en especial porque el usuario está casi obligado a contemplar todas y cada una de las posibilidades.

Quizá la característica más notable de una hoja electrónica para gestión empresarial ejecutada en una máquina adecuada, sea su asombrosa velocidad de funcionamiento. Ésta es una función de la programación en código de lenguaje máquina y no es nada sorprendente que la velocidad de un paquete escrito en BASIC para un pequeño ordenador personal pueda ser, en comparación, motivo de alarma.

Tal vez resulte interesante considerar algunos de los problemas con los que uno se podría enfrentar si intentara escribir un programa de este tipo en BASIC, aunque sólo fuera como un medio para llegar a comprender la complejidad que entrañaría dicha tarea.

Para empezar, cada celda se habría de definir de tres maneras. La celda debería ser capaz de retener datos de "variables alfanuméricas", como "enero" o "tarifas"; debería ser capaz de retener datos numéricos para su utilización en operaciones aritméticas, como por ejemplo, el total de las cuotas pagadas en enero; y también podría ser necesario que retuviera una fórmula que, en esencia, fuera una línea del código de programación, como "cuotas anuales/12" para obtener la cuota mensual. Luego las dimensiones de cada celda deberían ser susceptibles de ampliarse y reducirse, pero sin perder su parte menos significativa, para lo cual todas ellas deberían duplicarse: una aparecería en la pantalla y la otra, que siempre retendría toda la información, quedaría oculta dentro del programa.

Como se puede apreciar, sólo la manipulación de la información es en sí misma una tarea complicada; y recuerde que los paquetes más sofisticados ¡pueden constar de hasta 16 000 celdas individuales! Las técnicas que se utilizan para escribir esta clase de programas se parecen mucho a las que se emplean para escribir intérpretes para lenguajes como el BASIC o el FORTH, y también se hace uso de técnicas similares para el software de gestión por base de datos.

Todo esto viene al caso para explicar el porqué del elevado costo de estos programas empresariales escritos con una finalidad concreta. Un paquete como el Visicalc o el Supercalc puede ser muy caro, pero a la hora de contemplar la posibilidad de comprarlo, quizá el punto a considerar sea el ahorro que supone su utilización. La relativamente sencilla aplicación que realizamos anteriormente, la del director que recopilaba presupuestos, por ejemplo, puede representar un ahorro de quizá entre un 15 y un 20 % en un solo año. El costo del software constituye una parte insignificante en relación al monto total de inversión sin contar el ahorro de tiempo y energía que trae aparejado su aplicación. Por cierto, el paquete Visicalc probablemente haya sido el mejor amigo de los agentes de ventas del Apple.

Ventanas al mundo
Así como podemos desplazar el cursor por la pantalla hacia la derecha o hacia la izquierda, hacia arriba o hacia abajo, mediante el control del teclado, en una hoja electrónica podemos mover la pantalla a través de la hoja. Esta sofisticación permite visualizar una superficie mucho mayor que la de la pantalla.
Algunos ordenadores portátiles con pantalla de dimensiones limitadas, como el Epson HX-20 y el Osborne-1, utilizan el mismo sistema para todas sus operaciones

	A	B	C	D	E	F	G	H	I
1	TITULOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO
2	ALQUILER	180,00	180,00	180,00	192,00	192,00	192,00	192,00	192,00
3	TASAS							51,90	51,90
4	SEGURO								
5	TOTAL								--------
6	FAMILIA								243,90
7	TRANSPORTE								,00
8	COMIDA								,00
9	ROPA								,00
10	LIBROS								,00
11	TOTAL								--------
12	ESCUELA								,00
13	GASOLINA								92,00
14	REVISION								,00
15	OBRAS								18,00
16	IMPUESTOS								,00
17	SEGURO								,00
18	KILOMETRAJE								900,00
19	TOTAL								--------
20	COCHE								110,00
21	VARIOS								,06
22	VIAJES								74,00
23	ALOJAMIENTO								340,00
24	COMIDA								190,00
25	DIVERSIONES								38,00
26	TOTAL								--------
27	OCIO								642,06

```
  !   D    !!    E    !!    F    !!    G    !!    H   !
 8!     8.00      5.20      8.00      8.00      4.00
 9!     6.30      2.25       .00     18.00      3.40
10!      .00       .00     14.50       .00      6.95
11!---------------------------------------------------
12!    16.80      8.95     25.70     29.20     15.95
13!    34.00     41.00     48.00     35.00     39.00
14!      .00    112.00       .00       .00     44.00
15!      .00       .00       .00       .00       .00
16!      .00       .00       .00       .00       .00
17!      .00       .00       .00       .00    182.00
18!   600.00    950.00   1200.00    620.00    820.00
19!---------------------------------------------------
20!    34.00    153.00     48.00     35.00    265.00
21!      .06       .16       .04       .06       .32
22!      .00       .00       .00       .00       .00
23!      .00       .00       .00       .00       .00
24!    72.00     36.00     41.00     18.00     46.00
25!    13.00     22.00     14.00     13.00     22.00
26!---------------------------------------------------
27!    85.06     58.16     55.04     31.06     68.32
 H12          Form=H7+H8+H9+H10
Width:  9  Memory: 20 Last Col/Row:027    ? for HEL
 1?_
```

Juegos de aventuras

El ordenador puede ser una fuente de diversión. En un juego de aventuras, uno no es espectador: es el protagonista

Pronuncie la palabra "aventura" y la mayoría de la gente pensará en un libro, una película, un programa de televisión e incluso, quizá, en una experiencia personal. Pero muchas personas pensarán en los ordenadores, porque existe multitud de propietarios de ordenadores para quienes la palabra Aventura (con A mayúscula) se encuentra asociada a una clase de juegos por ordenador muy específica.

Para hacernos una idea de en qué consisten las aventuras por ordenador, comparémoslas con los libros. Cuando se lee una historia de aventuras se disfruta con los peligros, los misterios y los sucesos emocionantes; pero el protagonista es otra persona. En una aventura por ordenador, en cambio, uno no es un espectador, sino que forma parte de la acción. Como principal protagonista de la historia, se encuentra inmerso en la acción y es uno mismo quien está viviendo la experiencia.

En un libro, el lector no puede modificar el curso de los acontecimientos de la narración. Los sucesos siempre son los mismos y se producen en un orden determinado, y por más que se los vuelva a leer una y otra vez, nada variará en lo más mínimo. En cambio, en una aventura por ordenador sus decisiones, pareceres y acciones determinan la forma en que se desarrolle el argumento. Puede haber cualquier número de variaciones en cuanto al orden de los acontecimientos y muchos finales diferentes, algunos agradables y otros no tanto. En un juego de aventuras lo esencial es que uno es parte activa, aunque rodeado del confort y la seguridad de su propio hogar.

Cada aventura se desarrolla en un ambiente diferente. Éste puede ser un extraño mundo subterráneo, una feria al aire libre, una ciudad fantasma, otro planeta, una tierra mítica; en realidad, cualquier lugar. La acción se puede situar en el pasado, en el presente o en el futuro.

Por lo general la aventura posee una trama coherente que incluye un objetivo final. Por ejemplo, la meta puede ser huir de un planeta desconocido, encontrar y destruir a un genio del mal, rescatar a una princesa, hallar un tesoro o esclarecer un crimen y arrestar al culpable.

Por muy interesantes que le parezcan estos ingredientes, el verdadero placer de jugar a un juego de aventuras reside en solucionar los enigmas. Éstos constituyen una parte esencial de la aventura. Se puede encontrar un enigma en una de cuatro situaciones. La situación más común consiste en tener que resolverlo para poder seguir avanzando; por ejemplo, al hallarse ante el peligro de un puente a punto de derrumbarse. En otro sentido, el enigma suele ser una pista falsa: después de haber atravesado un desfiladero con el fin de abordar a la extraña figura que lo había estado observando desde la otra orilla, usted sólo encuentra ante sí un enorme espejo. Hay enigmas cuya resolución ayuda a completar el juego exitosamente, aunque no sean un factor esencial: por ejemplo, el descubrimiento de un pasaje secreto que conduce hasta una ciudad de enanos viciosos. Puede ser una cuestión de vida o muerte: usted se encuentra solo y abandonado en el saliente de una montaña, y no existe forma alguna de seguir ascendiendo o de retroceder, y no tiene ni comida ni bebida.

Los enigmas se resuelven utilizando el sentido común y no requieren ninguna experiencia ni conocimientos especiales. No obstante, el aventurero ha de estar alerta, ya que las claves para la solución siempre se han de hallar en el texto o bien se deben deducir del mismo. En los juegos de aventuras bien escritos no existen elementos dejados al azar, excepto en dosis muy pequeñas.



Ian Dobbie

Un papel preponderante
Los juegos de aventuras, como *Dungeons and dragons* (Calabozos y dragones), existen ya desde hace muchos años; pero en la actualidad el ordenador personal los ha puesto al alcance de la mano de una gama de jugadores potenciales mucho más amplia. Los juegos de aventuras empiezan a competir, en cuanto a popularidad, con los juegos recreativos. Aunque la mayoría de ellos aún son para un solo jugador, pueden ser una absorbente fuente de recreación para todos los miembros de la familia. El jugador asume el papel de protagonista

Al jugar a una aventura probablemente encontrará objetos, mensajes y personajes que parecerán irrelevantes en el contexto de la historia. No obstante, siempre debe tener presente que en esta clase de juegos prácticamente todo tiene un fin determinado, incluso aunque, en ocasiones, esa finalidad específica sea la de despistarlo. ¿Qué significado puede tener un montón de botellas de ron rotas? ¿Cómo debe uno interpretar una voz profunda que le dice «REPITA»? ¿Qué utilidad práctica puede ofrecer un lodazal maloliente? ¿Por qué ese felpudo estaba clavado al suelo si no había nada debajo de él? Todos estos enigmas están incluidos en varios juegos de aventuras y desempeñan un papel esencial en la trama. Cuando se encuentra un objeto por primera vez, independientemente de lo común o extraño que resulte, es muy improbable que

se tenga de inmediato la certeza de su importancia; sin embargo, es seguro, como dos y dos son cuatro, que el jugador lo necesitará mucho antes de lo que cree.

Muchos juegos de aventuras poseen un pequeño laberinto en el que cada sitio o habitación se describe en idénticos términos. La única manera fiable de trazar un mapa de ruta a través de este tipo de laberintos, consiste en imitar a Hansel y Gretel, es decir, dejar caer pequeños objetos en el sendero que se sigue, con el fin de poder identificar cada cuarto. Esta estratagema se ha hecho tan conocida que algunos programadores han ideado problemas adicionales; por ejemplo, que un ladrón vaya tras el aventurero y, con la mayor discreción, cambie la disposición de los objetos que sirven de orientación.

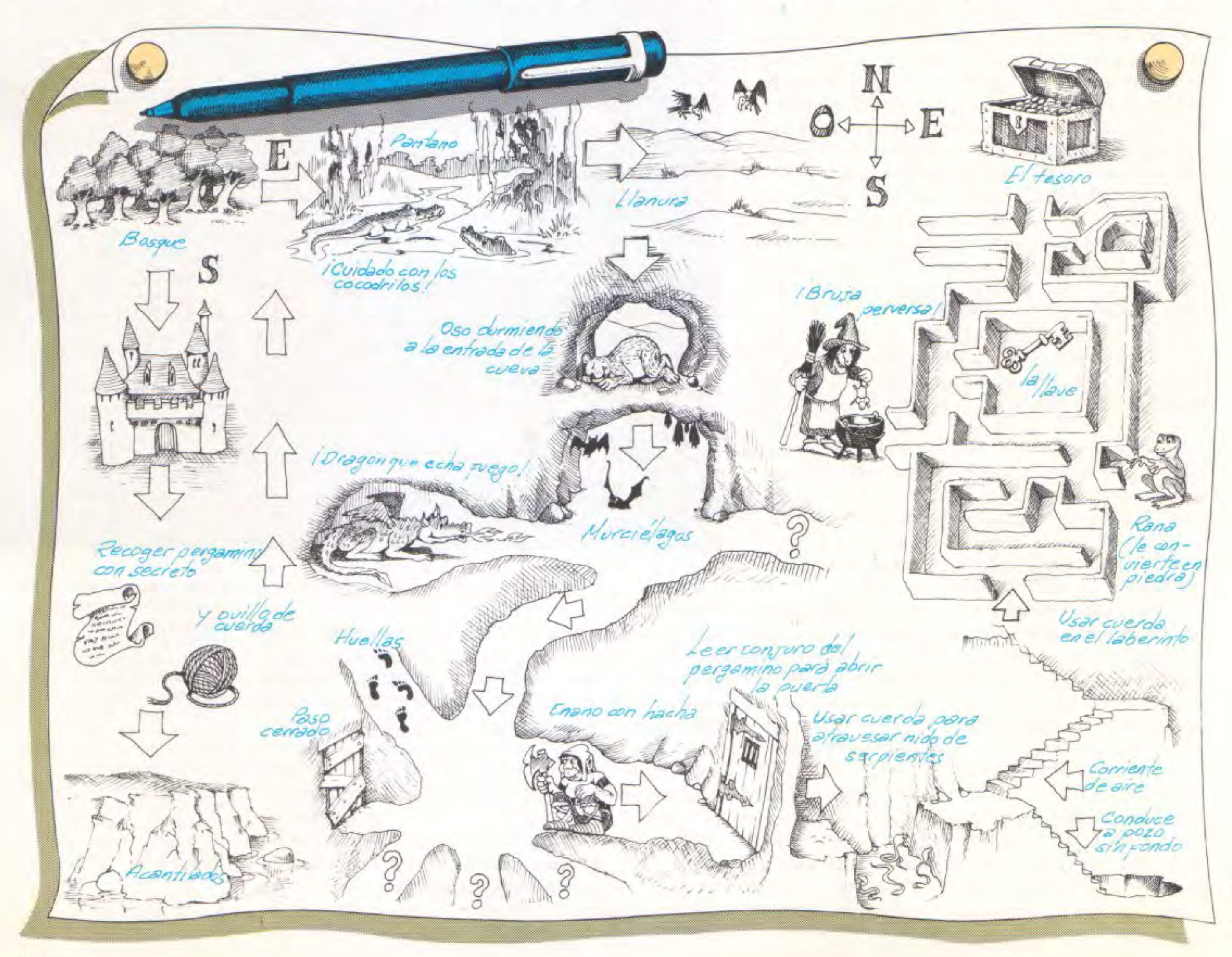
En algunas aventuras, si bien para completar el juego exitosamente se han de resolver todos los enigmas y superar todos los escollos, el orden en que se resuelvan los misterios y se alcancen los objetivos carece de importancia. Este tipo de juegos se opone a aquellos en los cuales existe un único camino para llegar a un desenlace triunfal.

Una buena aventura puede ocuparle horas, incluso semanas de juego hasta que se descubren todos sus secretos. Entonces, cuando decida suspender el juego, estará capacitado para guardarlo en la fase en que se encuentre, en una cassette o en un disco. Este recurso también resulta útil cuando se ha llegado a una etapa peligrosa de la aventura, porque continuar avanzando impertérrito a la deriva hacia un grupo de orcas, provisto apenas de una lámpara y una botella de agua, es proceder con excesiva temeridad. Un aventurero prudente "congelará" el juego antes de tener que vérselas con las orcas. Luego, si éstas deciden terminar con los días del aventurero, por lo menos existirá la posibilidad de reanudar el juego después e intentar un curso de acción diferente, lo cual resultará mucho menos desalentador que volver a empezar la aventura desde el principio.

Incluso aunque el jugador sea muerto, ello no implica necesariamente un final definitivo. Algunos autores permiten que el difunto resucite, a menudo en medio de una humareda color naranja, a veces perdiendo puntos o algunas de sus posesiones, y siempre apareciendo en un lugar más bien indeseable, como, por ejemplo, el infierno o en medio de la nada.

¿Cómo participar y comunicarse con el programa? Éste puede dirigirse directamente al jugador o bien éste puede ser representado por un "muñeco", personaje al que se controla mediante órdenes. El ordenador actúa al mismo tiempo como intérprete de los deseos del aventurero y como narrador. El jugador da entrada a las órdenes a través del teclado y éstas se visualizan en la pantalla.

Algunas aventuras visualizan en pantalla sólo el texto, otras únicamente gráficos, el resto, una mezcla



David Higham

Lectura de un mapa
En un juego de aventuras, el jugador o el personaje que éste representa se ha de mover en un vasto territorio o ambiente; es muy corriente que exista una red de pasadizos subterráneos. Algunos fabricantes incluyen pistas en los manuales de instrucción para aquellos jugadores que, llegado un momento, se queden absolutamente atascados; pero la única forma de descubrir el trazado consiste en abrirse camino a través de él. Para completar un juego de este tipo se pueden invertir semanas enteras, de manera que es imprescindible llevar un boceto esquemático que le indique en qué lugares ha estado y los obstáculos que ha encontrado a su paso

de ambos. Los efectos sonoros se utilizan por lo general en los juegos de aventuras exclusivamente con gráficos. Las aventuras de texto se pueden comparar con un libro sin ilustraciones, en el cual los lugares, objetos y acontecimientos se describen mediante palabras. En la mayor parte de los juegos que combinan texto y gráficos, éstos sirven de complemento a las descripciones del texto y a menudo son escenas estáticas de objetos y lugares. Estas escenas pueden ser desde unos sencillos trazados lineales hasta imágenes muy detalladas. En las aventuras con gráficos, éstos normalmente se utilizan en mapas estilizados o imágenes del terreno, así como para representar los interiores de las construcciones. Los personajes y los objetos se representan mediante símbolos o figuras en miniatura. En este caso, la acción del jugador por lo general queda limitada a una pequeña cantidad de órdenes, básicamente pulsaciones individuales de teclas para hacer mover y controlar al personaje.

La visualización de texto en una aventura comprende, por lo común, tres elementos: dónde se halla el jugador, qué puede ver y a dónde puede ir. Por ejemplo, el texto en pantalla puede decir: «Usted se encuentra en la espesura de un bosque. Sobre su cabeza, el cielo no es más que una serie de manchas entre el denso follaje. Un sendero muy transitado lleva hacia el este y hacia el oeste. Delante, hacia el norte, se abre un abismo. En el suelo está clavada una espada; enroscada en ella hay una serpiente verde». Al jugador se le describen los alrededores, se le dan algunas de las direcciones que puede tomar y se le indican los objetos que están a la vista.

Las órdenes constan, por lo general, de dos palabras, un verbo seguido de un sustantivo, si bien las aventuras más sofisticadas incluyen oraciones completas. Algunos de los verbos estándar son GET (coger), DROP (soltar), PUSH (empujar), PULL (tirar), THROW (arrojar), LIGHT (encender), KILL (matar), EAT (comer) y DRINK (beber). GO NORTH (ir hacia el norte), por ejemplo, sería la forma normal de especificar un movimiento, si bien la mayoría de los juegos de aventuras permiten abreviar este tipo de órdenes, y así, es muy corriente el empleo de iniciales, como W para GO WEST (ir hacia el oeste).

EXAMINE (examinar) es un verbo esencial; a menudo es el medio para adquirir mayor información. En el ejemplo anterior, EXAMINE SNAKE (examine la serpiente) podría significar: "Se trata de una culebra de hierba", o tal vez: "La serpiente advierte que usted avanza hacia ella y lo ataca". Algunos verbos no requieren un sustantivo.

INVENTORY (inventario) se emplea para decirle lo que está llevando. Algunos objetos se pueden colocar dentro de otros (agua en una botella o un hacha en un saco, por ejemplo), mientras que otros se pueden llevar encima (por ejemplo, un anillo o una capa).

Con frecuencia se emplea SCORE (puntuación) para que el jugador sepa qué progresos ha hecho en alcanzar los objetivos. Digitar HELP (ayuda) puede facilitar una pista para superar una dificultad, pero generalmente le aconseja que siga intentándolo. En ocasiones, en respuesta a una orden determinada, se le puede decir al aventurero: "¡No puede hacer eso... todavía!", lo que implica que una combinación verbo-sustantivo dará un resultado concreto pero no en ese momento y, quizá, no en ese lugar. Una parte del desafío consiste en averiguar los verbos y los sustantivos que son relevantes para la aventura. Las palabras y las combinaciones que ésta no reconoce suelen dar como respuesta algo como "No le comprendo".









Ian McKinnell

En busca del tesoro

El Hobbit es un juego de aventuras con gráficos para el Sinclair Spectrum y toma su nombre de la novela de J. R. R. Tolkien, un ejemplar de la cual se proporciona con la cassette. Usted asume el papel de Bilbo, viajando a través de Middle Earth y encontrándose con muchos de los personajes y situaciones del libro, en busca del dragón y su tesoro

La mayoría de los editores de aventuras facilitan hojas de pistas para quienes se quedan trabados, ofreciendo sutiles claves para seguir adelante.

Algunos juegos de aventuras son demasiado largos para la memoria del ordenador. Para superar este inconveniente se proporcionan en disco; el programa principal se carga en la memoria al principio y los textos y gráficos se toman del disco o se devuelven a él a medida que se los va necesitando. Pero gracias a las inteligentes técnicas para comprimir grandes cantidades de texto en una memoria restringida, ahora un juego de aventuras largo se puede almacenar en la memoria de un ordenador, lo que significa que estas aventuras se pueden suministrar en cinta de cassette.

Existen aventuras para virtualmente toda clase de micros. Cualquiera le proporcionará una grata experiencia y puede ser el comienzo de un hobby.

¡Felices aventuras!



Ian Dobbie

Resolviendo un caso
Deadline (Plazo fatal) es una variante temática de las aventuras. Usted encarna el papel de un detective que debe esclarecer un asesinato. Su archivo contiene, por ejemplo, los resultados de la autopsia, algunas píldoras que se hallaron junto al cuerpo y otras notas relativas al caso. El juego no se desarrolla en tiempo real (dos semanas se considera un muy buen tiempo), pero cada acción que usted empreda, como ir de un cuarto a otro o interrogar a un sospechoso, representa unos minutos vitales que se descuentan del tiempo límite de 12 horas que se le ha dado para resolver el caso

Micromundos

La mayoría de los programas educativos con ordenadores no son sino libros de texto electrónicos. El LOGO es diferente: se vale de un ordenador para crear un "entorno de aprendizaje"

La tortuga que camina
La tortuga se diseñó como un dispositivo con el cual pensar, particularmente durante el ciclo de aprendizaje de geometría y relaciones espaciales. Cuando los niños no están seguros acerca de cómo instruir a la tortuga para que realice una maniobra determinada, tienden a asumir el papel de la tortuga y gatear por el suelo obedeciendo las instrucciones en LOGO. Esto convierte el aprendizaje en una experiencia mucho más "real"


Ian McKinnell

Desde que el micro apareciera por primera vez, en 1977, los educadores con amplitud de miras comprendieron rápidamente el enorme potencial que ofrecía como medio auxiliar de enseñanza en la escuela. En la actualidad la mayoría de las escuelas de los países desarrollados poseen al menos una máquina y en muchas de ellas la informática es una materia más de estudio. A pesar de ello, la incidencia del micro en los métodos de enseñanza tradicionales ha sido ínfima.

La prueba más evidente a este respecto es la gama de programas educativos para ordenadores personales que se ofrece comúnmente, que, en líneas generales, denota una notable falta de imaginación. La mayor parte de estos programas se pueden describir como "libros de texto electrónicos", en los cuales el ordenador le presenta al alumno una serie de "cuadros" en la pantalla (equivalentes a las páginas de un libro de texto) y luego comprueba en qué medida el alumno ha asimilado la información, a través de una serie de preguntas tipo test que deben ser contestadas mediante la elección de una entre varias respuestas, que el ordenador califica automáticamente sin la intervención del profesor.

Los paquetes de programas de este tipo son muy fáciles de escribir en un ordenador personal y ofrecen la ventaja de acompañar el texto con gráficos a todo color (y, en ciertos casos, también animados). Sin embargo, esto no es más que una automatización del método tradicional, y no una forma original de aplicar el potencial del microordenador.

El lenguaje LOGO presenta una perspectiva diferente. A partir de la obra del profesor Seymour Papert, del Massachusetts Institute of Technology, el LOGO se define como "una filosofía de educación y una familia de lenguajes de programación para ordenadores concebida para ayudar a la realización de esa filosofía".

Muchas personas han considerado erróneamente al LOGO como un simple lenguaje de programación y han comparado sus órdenes y configuraciones con las del BASIC, llegando a la conclusión de que el LOGO es un lenguaje mucho mejor para los principiantes. Este razonamiento se aparta de la verdadera cuestión. Papert jamás tuvo la intención de crear un sistema destinado a enseñar al niño a programar. Él lo concibió como un entorno donde el alumno pudiera aprender diversos temas, un ambiente en el que, en realidad, pudiera descubrir una manera de aprender.

Gran parte de esta filosofía deriva del eminente filósofo suizo de la educación Jean Piaget, quien afirmó que el niño, en un entorno adecuado, era capaz de aprender cualquier tema por sí mismo del mismo modo que aprende a andar y a hablar. Sin embargo, la obra de Piaget era completamente teórica, y Papert se abocó a la tarea de crear un ambiente práctico para las teorías de aprendizaje de Piaget.

Que los métodos de educación tradicionales no cumplen este objetivo se hace evidente a partir del hecho de que la mayoría de los adultos teme aprender y no disfruta con la idea de tener que abarcar nuevas áreas de conocimiento. En su libro *Mindstorms — children, computers and powerful ideas (Confusión mental: niños, ordenadores y conceptos eficaces)*. Papert afirma que el síntoma más común de esta actitud del adulto es el miedo generalizado a las matemáticas o, como dice él, la "matematofobia".

Una de las causas de esta postura reside en que la mayor parte de estas disciplinas se imparten de la misma forma, en tanto que sus aplicaciones son totalmente diferentes. Al niño se le enseña, por ejemplo, a multiplicar en la misma forma en que aprende las capitales del mundo: mecánicamente. El proceso de aprendizaje se divorcia de lo que se está aprendiendo, cuando ambas cosas deberían ser inseparables.

Para el propio Papert, el aprendizaje de un conocimiento nuevo, ya sea volar, cocinar o aprender un idioma extranjero, es como practicar una afición. Él atribuye esta actitud suya a su infancia, cuando descubrió, a muy temprana edad, cómo funcionaban las ruedas dentadas y aplicaba este concepto cada vez que se enfrentaba a un nuevo problema. También Albert Einstein solía decir que cuando se encontraba con algo que no comprendía, lo descomponía en conceptos que había aprendido antes de los cinco años.

Estas eficaces ideas se incorporaron al LOGO, como se puede ver en nuestro ejemplo del LOGO en acción. La primera característica importante de este lenguaje es la tortuga, que fue diseñada como "un dispositivo con el que pensar" del mismo modo en que Papert usaba las ruedas dentadas cuando era un niño. Para los niños pequeños, la tortuga asume la forma de un robot móvil diseñado especialmente (véase p. 176), que está conectado a un micro y que se puede desplazar por el suelo digitando órdenes en LOGO. Normalmente la tortuga lleva un lápiz para dibujar formas en el suelo y también puede estar dotada de un pequeño altavoz y de detectores de colisión para guiarla a través de un camino con obstáculos.

Los niños, por lo general, luego de emplear las tortugas móviles, pasan a utilizar las tortugas de pantalla, formas que se pueden mover a través de la pantalla del

Aprendiendo la curva

Este ejemplo ficticio pero típico muestra cómo el LOGO estimula a un grupo de niños a resolver problemas con los cuales nunca antes se habían encontrado.

```
TO CURVE
REPEAT 80
  FORWARD 1
  RIGHT 1
END
CURVE
```

—Si queremos pétalos necesitamos dibujar una curva.
—Pero la tortuga siempre se mueve en línea recta.
—¿Y qué pasaría si la obligáramos a desplazarse una distancia corta, luego la hiciéramos girar sólo un poquito, luego otra vez en línea recta, y así varias veces? Eso sería como una curva.
—De acuerdo; la distancia más pequeña es uno, y el ángulo más pequeño es uno. Hagámosla hacer eso ochenta veces.

```
CURVE
CURVE
```

—¡Eh!, eso es justo lo que queríamos.
—Dos de ellas harán un pétalo; probemos.

```
TO PETAL
CURVE
RIGHT 90
CURVE
END
PETAL
```

—Eso no tiene nada que ver con un pétalo. ¿Qué pasó?
—Que siguió a partir de la última curva: le deberíamos haber dicho que fuera en otra dirección.
—Pero, ¿cuánto la hacemos girar?
—Probemos con noventa; eso suele dar resultado.
—Y hagamos una palabra nueva: PETAL (pétalo), para ahorrar tiempo.

```
TO PETAL
CURVE
RIGHT 100
CURVE
END
PETAL
```

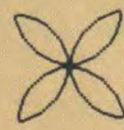
—Eso está mejor, pero noventa grados no era suficiente. ¿Qué hacemos ahora?
—Pensemos algo y veamos si funciona, en vez de hacer suposiciones.
—Sí; recuerda que aprendimos que si la tortuga gira hacia la derecha alrededor de algo gira a través de un total de trescientos sesenta grados.
—Bien, sabemos que gira ochenta en la primera curva; entonces tiene que girar ochenta en el camino de vuelta, o sea ciento sesenta grados.
—Dejando doscientos para girar en la punta del pétalo.
—No, porque para volver a la posición desde donde empezó, necesitará girar también en la otra punta.
—Entonces deberíamos probar con la mitad de doscientos.

```
PETAL
PETAL
PETAL
PETAL
```



—Magnífico: con cuatro como éste haremos una flor.

```
TO FLOWER
REPEAT 4
  PETAL
  RIGHT 10
END
FLOWER
```



—No salió muy bien que digamos; está dispareja.
—Nos olvidamos de poner algunos giros entre los pétalos.
—Pero entonces, ¿por qué no los dibujó uno encima del otro?
—Porque después de dibujar un pétalo, la tortuga queda mirando cien a la izquierda de donde empezó originalmente.
—De modo que cada pétalo gira cien hacia la izquierda cada vez.
—Eso está bastante bien: lo que necesitamos es noventa, así que agreguemos un giro de diez hacia la derecha entre cada pétalo.

```
FLOWER
RIGHT 180
FORWARD 100
RIGHT 180
PETAL
```



—¡Al fin! Gira la tortuga para que podamos dibujar el tallo.
—Cien debería ser suficiente.
—Pongamos una hoja en la punta; puede ser igual a un pétalo.
—Pero esta vez recuerda el ángulo: la tortuga ha de dar vuelta hacia la derecha.

Liz Dixon

ordenador. La tortuga es un eficaz dispositivo con el cual los niños pueden aprender los conceptos básicos de las relaciones espaciales, que les permitirán luego iniciarse en la geometría avanzada.

El control de la tortuga es, sin embargo, sólo una pequeña aplicación del LOGO, pero es el aspecto más promocionado a nivel publicitario porque visualmente es el más interesante. Más importante es el concepto de establecer, a partir de ideas sencillas, ideas más sofisticadas y, a la inversa, descomponer grandes problemas en problemas más pequeños del tipo de los que ya se han asimilado previamente.

Estos procesos se pueden ver con toda claridad a través de la conversación imaginaria entre un grupo de niños que están aprendiendo a instruir a la tortuga para que dibuje una flor (véase recuadro). Comienzan con sólo tres órdenes disponibles: FORWARD (adelante), que hace que la tortuga se mueva hacia adelante hasta una distancia calculada; RIGHT (derecha), que obliga a la tortuga a describir un ángulo determinado, y, por último, REPEAT (repetir), que repite una cierta cantidad de veces las líneas que hay instrumentadas en el programa.

A partir de estas ideas fundamentales los niños construyen primero una "herramienta" (un programa) para dibujar una curva (TO CURVE... END). Toda esta secuencia se puede producir ahora simplemente digitando CURVE. Siguiendo el mismo proceso, después de experimentar y de un aprendizaje posterior, se define una orden PETAL (pétalo), que se vale de la orden CURVE. Finalmente se desarrolla una orden FLOWER (flor), que dibujará la imagen completa.

El LOGO no es el único lenguaje que incorpora este tipo de estructuras (el FORTH es otro de ellos, véase p. 150), pero es el único diseñado para que lo utilicen niños pequeños. Prescinde de muchas de las formalidades y procedimientos asociados con la programación en otros lenguajes. De hecho, se trataba de que el niño no fuera consciente de estar programando un ordenador, sino sólo de estar resolviendo un problema.

En algunas situaciones de aprendizaje, el alumno no se confunde ni siquiera a este nivel. El maestro establece una serie de eficaces herramientas utilizando el LOGO, todas las cuales se refieren a un tema o área de conocimiento determinados. Al niño se le permite luego explorar el tema empleando las herramientas y descubrirlo por sí mismo. Estas áreas se denominan "micromundo": entornos limitados en los cuales se utiliza el ordenador para imitar algo del mundo real o alguna área de conocimiento.

Probablemente el mejor ejemplo de lo que es un micromundo sea el modelo en LOGO de la física de Newton. Aunque la primera ley de Newton dice que sin la influencia de fuerzas externas un cuerpo se seguirá moviendo en línea recta a una velocidad constante, las mentes de los pequeños observan que en el mundo real todo va moviéndose más despacio. Esto determina que el aprendizaje se bloquee. No obstante, utilizando el LOGO se puede construir un micromundo en el que todo se comporte de acuerdo con las leyes de Newton y, con la ayuda de herramientas con las cuales se empujan los objetos a través de la pantalla, los niños aprenden rápidamente y por sí mismos las tres leyes de Newton.

El LOGO es un concepto poderoso que vale la pena aprender mediante un ordenador personal. Las tortugas móviles no son baratas. Están comenzando a salir al mercado versiones de LOGO que emplean tortugas de pantalla para diversos ordenadores personales.

La era de los portátiles

A medida que aumenta la sofisticación de los ordenadores, se los puede comprimir más, convirtiendo en realidad el micro portátil



Ian Dobbie

No es un juguete para ejecutivos
Los ordenadores portátiles como el Epson HX-20, con hasta 32 Kbytes de memoria, periféricos incorporados y una amplia gama de software, han aumentado de manera considerable la cantidad de información de que puede disponer un ejecutivo de empresa, independientemente de en qué lugar se halle. Cada microcassette puede retener hasta 230 Kbytes (quizá 40 000 palabras, equivalente a una amplia base de datos), mientras que los programas estándar de tratamiento de textos y hoja electrónica le permiten llevar a cabo su trabajo casi en cualquier parte: ¡hasta en un taxi!

El desarrollo del micro portátil, o nanocomputador, tal como lo conocemos hoy en día, provino de una doble vertiente. Una de éstas fue el aumento de tamaño de las calculadoras de bolsillo, como en el caso de la Sharp PC1251 y la Casio FX700P. La otra fue un proceso evolutivo de miniaturización, a partir del cual se produjeron máquinas como los ordenadores personales Tandy TRS80 Modelo 100 y el Epson HX-20 (véase página 169).

Estos adelantos fueron consecuencia directa del desarrollo de chips comprimidos en mayor medida, que permitieron incluir muchísima más información en el mismo espacio físico.

Con el advenimiento del microprocesador de un solo chip, en 1972, se hizo teóricamente posible construir un ordenador completo en una caja de tamaño no mayor que el de un paquete de cigarrillos. Sin embargo, las dimensiones de la pantalla de visualización y el principal medio de acceso (el teclado) impusieron limitaciones prácticas a esta miniaturización.

No obstante, las calculadoras de bolsillo se fueron haciendo más pequeñas y en la actualidad existen relojes digitales que funcionan también como calculadoras, para lo cual requieren un puntero o una aguja incorporada con esa finalidad para operar las "teclas". Pero en ellos resulta bastante difícil realizar una sencilla operación aritmética, e incluso el más entusiasta renunciaría a la tarea de dar entrada a un programa de 50 líneas en BASIC. De manera que es poco probable que los ordenadores personales lleguen a ser tan pequeños como un reloj de pulsera. Pero han llegado a ser del tamaño de una calculadora.

Una vez que las calculadoras de bolsillo se hicieron programables, si bien en su propia notación de programación, faltaba sólo un paso para la incorporación de un lenguaje de alto nivel, y el más adecuado era el BASIC. Casi simultáneamente, los fabricantes comenzaban a utilizar una RAM "estable" (una clase de memoria que, reteniendo una pequeña carga eléctrica, no pierde su contenido al interrumpirse la alimentación eléctrica) y un generador de caracteres más amplio para permitir la visualización de caracteres alfabéticos además de los numéricos.

Dispositivos de precio razonable, como los ordenadores de bolsillo Sharp y Casio, que poseen una gama de órdenes en BASIC comparable a la de los ordenadores personales y capacidad de memoria similar, están invadiendo rápidamente el mercado de las calculadoras programables. Diseñados para que se puedan llevar cómodamente en el bolsillo (incluso en el de una camisa), algunos ofrecen la posibilidad de poderlos conectar de manera directa, mediante interfaces, tanto con una impresora como con una grabadora de cassette. Sin duda, en el caso de que exista suficiente demanda, el próximo paso será incorporar un dispositivo para comunicaciones que permita transferir información a través de las líneas telefónicas. La gama Casio, sobre todo, resulta muy conveniente para científicos e ingenieros, ya que retiene la amplia gama de funciones matemáticas y científicas que hicieron tan atractivas las calculadoras de esta marca.

Existen modelos con impresora incorporada que incluyen interface para cassette, como el PC1251 de Sharp.

El siguiente paso trascendental lo representan nanocomputadores como el Epson HX-20, el Tandy 100 y el NEC 8220. Estos micros ofrecen una completa versión de BASIC (Microsoft en todos los casos), una RAM utilizable de entre 16 y 64 Kbytes, una LCD (*Liquid Crystal Display*: visualización en cristal líquido) incorporada de dimensiones razonables (20 × 4 caracteres en el Epson, 40 × 8 en los otros dos) y la posibilidad de conectarlos a una amplia gama de periféricos estandarizados a nivel industrial.

Por cierto, si se los acopla a un monitor regular (para ello quizá sea necesario adquirir una interface especial), se tendrá un micro comparable, en cuanto a capacidad y rendimiento, a un ordenador personal convencional y al mismo precio. La diferencia fundamental radica en el hecho de que estas máquinas dependen de su propia fuente de alimentación eléctrica interna, lo cual les confiere una absoluta portabilidad.

El Epson, por ejemplo, posee pilas de cadmioníquel incorporadas de las cuales una carga completa dura hasta 50 horas, y el sistema está diseñado de tal manera que cuando las pilas se están agotando él

mismo se interrumpe para preservar la información esencial.

En comparación, el NEC y el Tandy utilizan un CMOS (*Complementary Metal Oxide Semiconductor*: semiconductor de óxido metálico complementario), que requiere mucha menos energía y les permite funcionar con pilas normales tan ligeras como un lápiz.

Al Epson se le consideró en un principio como una "ayuda para ejecutivos", pero es más adecuado para utilizarlo fuera de la oficina que dentro de ella. Es perfectamente apropiado para captación de datos (recolección de información) en la fábrica, en lugares alejados o incluso mientras se camina por la calle, para ser procesados luego en la oficina.

En la práctica el Epson ofrece diversas maneras de transmitir datos. Usado con accesorios patentados, se puede emplear como un terminal de télex; a través de un modem es posible conectarlo a otra máquina vía línea telefónica; o, lo que resulta más sencillo y más barato, los datos se pueden almacenar en cinta de cassette y enviarlos luego por correo.

Es precisamente esta adaptabilidad y versatilidad lo que hace que estas máquinas les resulten tan atractivas a personas que nunca antes habían considerado la posibilidad de emplear ordenadores en su trabajo cotidiano. Un vendedor, por ejemplo, que realice veinte o más visitas al día simplemente para recoger los pedidos habituales, en vez de rellenar formularios impresos puede utilizar una máquina portátil de esta capacidad y potencial. El programa para procesar los pedidos le proporcionaría una lista de productos y cuando el cliente le hiciera su encargo él sólo apuntaría las cantidades. Al final de cada visita la información del pedido se volcaría en la cinta y, concluida la jornada, el vendedor enviaría la cinta por correo hacia la oficina central o bien transmitiría la información directamente a través del sistema informático principal de la empresa.

La propia impresora del micro le proporcionaría al vendedor una copia de sus pedidos del día y al cliente una confirmación inmediata.

Versiones ligeramente más sofisticadas pueden incluir, asimismo, los niveles de stock y advertir cuando éstos disminuyen peligrosamente, si bien en el caso de que varios vendedores trabajaran al mismo tiempo, se habría de conectar el micro con el ordenador principal durante cada visita. Esto no representaría ningún problema si se dispusiera de un acoplador acústico barato y liviano. En este punto nuestro portátil barato se convierte en un terminal interactivo, que permite que se pueda interrogar a toda la base de datos de la empresa obteniendo una respuesta inmediata.

Aparte de las evidentes ventajas que ofrece el acceso inmediato a la información actualizada, el ahorro de tiempo y costo que representa esta clase de entrada de datos en líneas puede amortizar en cuestión de meses el desembolso que supone un sistema portátil completo.

Así como los ordenadores de bolsillo más pequeños están desplazando a las calculadoras personales, los nanocomputadores se están desmarcando de los dispositivos de captura de datos. Estos "terminales mudos" a pilas (que no se pueden programar) han estado a la venta durante cierto tiempo en algunos países, pero no han llegado a popularizarse debido a su elevado precio y a las dificultades que entraña su funcionamiento.

Dados su adecuado software, su capacidad para conectarse a impresoras y monitores de oficina y, quizá, a unidades de disco o cintas de cassette con un acceso más rápido, probablemente los ordenadores portátiles sí se popularizarán. Las máquinas de este tipo proporcionan el primer mercado real para la memoria de burbuja: una memoria estable y autocontenida basada en un chip radicalmente distinto, que en un pequeño espacio ofrece una enorme capacidad de almacenamiento. Aplicaciones clásicas, como el Sharp PC5000, proporcionan 128 Kbytes de almacenamiento por cada cartucho enchufable, con un tiempo de acceso más rápido que el del disco. Resulta muy atractiva la posibilidad de unas máquinas pequeñas que posean un millón de bytes de memoria incorporados.

Esta capacidad de almacenamiento se encuentra disponible en disco en un tercer tipo de máquinas portátiles, representadas por el Osborne Executive, el Ajile Hyperion y el Portico Miracle. Mucho más caros que el Epson HX-20 o el TRS100, estos ordenadores pueden describirse con mucha propiedad como transportables, porque requieren enchufarse a un tomacorrientes. Existen paquetes de pilas para algunas de estas máquinas, pero es muy raro que superen las dos horas de vida útil.

La especificación de estos ordenadores suele incluir unidades gemelas de discos flexibles, monitor de televisión, teclado desmontable y un sistema operativo como el CP/M (*Control Program for Microprocessors*: programa de control para microprocesadores).



Ian McKinnell

Máquinas para todos los gustos
(En el sentido de las agujas del reloj, desde arriba, derecha.) El Epson HX-20, primero de la nueva generación de ordenadores portátiles, ofrece hasta 32 Kbytes de RAM y una amplia gama de periféricos. Con menos potencia (y menor precio) figuran el Sharp PC1251, que aparece con impresora y unidad de microcassette, y el Casio FX700P, que utiliza una grabadora de cassette estándar. Después de haber alcanzado un buen lugar en el mercado de ordenadores "transportables", el Executive, segundo modelo de la Osborne Computer Company, ofrece una pantalla mayor y otros refinamientos



Ian McKinnell

Un procesador de textos
El concepto de un "cuaderno de notas electrónico" ha dado un paso más hacia adelante con la introducción del *microwriter*, un procesador de textos que se puede manejar con una sola mano. Las seis teclas se pulsan en diferentes combinaciones para obtener el alfabeto completo. La salida puede ser directa a una impresora, con algunas órdenes de formato para que el usuario componga el texto, o bien a un procesador de textos de escritorio para su almacenamiento



Herramientas para el comercio
Los ordenadores portátiles baratos han encontrado su sitio en áreas anteriormente inaccesibles para los sistemas de procesamiento de la información

Los periodistas pueden dar entrada en su máquina portátil a las preguntas que desean formular durante las entrevistas, introducir junto a ellas las respuestas y luego editar el artículo, tal vez en el tren, rumbo a casa

Los farmacéuticos disponen de un modo sencillo y barato de etiquetar sus preparados, que también facilita la tarea de controlar las existencias

Los vendedores pueden hacer presupuestos a simple vista, incluso para cálculos complicados, como la instalación de un sistema de calefacción central o una póliza de seguro de vida

El control de existencias es muy importante aun en las tiendas pequeñas, pero en la industria, donde los valores de stock implican muchos millones de pesetas, resulta imprescindible mantener un registro exacto. Los micros portátiles, con 32 o 64 Kbytes de RAM, almacenamiento en cinta de cassette y configuración para comunicaciones, resultan ideales para este fin

David Higham

El primero que salió al mercado británico fue el modelo 1 de Adam Osborne, que ofrecía una gama de software completa, incluyendo BASIC, Wordstar y Supercalc y programas de CP/M por menos de 2 000 libras (unas 450 000 pesetas).

Su punto débil era la pantalla, demasiado reducida (4 × 3,5 pulgadas, o sea, 100 × 85 mm), por lo cual los caracteres eran más bien pequeños y hacía necesaria la utilización del movimiento de pantalla horizontal para conseguir un tamaño legible de visualización cuando se empleaba un paquete de tratamiento de textos como el Wordstar. Sin embargo, los usuarios se acostumbraron a ella rápidamente y el Osborne 1 se convirtió en líder del mercado británico.

La competencia se apresuró a ofrecer más configuraciones, especialmente en términos de potencia y velocidad del microprocesador, y de la capacidad de almacenamiento en disco, hasta que esta clase de máquinas llegó a resultar imposible de distinguir, excepto en su aspecto, de los sofisticados sistemas para oficinas y empresas. En muchos casos ofrecen compatibilidad absoluta con máquinas específicas, en particular el ordenador personal IBM, lo cual quizá favorezca la utilización de ellas como un "segundo ordenador".

La verdadera diferencia entre estos portátiles y los sistemas para oficina radica en su estructura. Un programa de aplicaciones para ejecutar en un sistema normal para oficina, podrá realizarse en un portátil que posea un microprocesador similar. La única limitación continúa siendo, probablemente, el tamaño de la pantalla del monitor y, en consecuencia, de los caracteres individuales.

Todo esto ha sido posible gracias al desarrollo del microprocesador de un solo chip. Las limitaciones en cuanto al nivel de acceso de entrada y salida son consecuencia de nuestra dependencia del teclado como principal dispositivo de entrada y del tubo de rayos catódicos como principal dispositivo de salida. Pero, ¿y si sugiriésemos el reconocimiento y la síntesis de voz como una alternativa? No existiría entonces razón alguna por la cual un micro de potencia similar a la del Ajile Hyperion, por ejemplo, pudiera ocupar más espacio que un Sony Walkman; y, con toda probabilidad, si dispusiéramos de la tecnología adecuada, tendría exactamente ese aspecto, con la incorporación de un micrófono en miniatura.

En el momento actual, cuando los avances tecnológicos se están sucediendo a tanta velocidad, es probable que se produzcan desarrollos en este campo, incluso aunque no incurramos en la fantasía de imaginar la entrada y salida de voz humana. Imaginemos, por ejemplo, una combinación entre el Sinclair ZX Spectrum, el televisor de pantalla plana de Sinclair y un microdisco, proporcionándole energía al sistema a través de una pila recargable de cadmio-níquel: ¿por qué no podemos juntar todo esto? Aunque quizá sea la Sinclair la que lo haga por nosotros...

Debido a su pequeño tamaño, el ZX Spectrum es una máquina muy adecuada para esta aplicación, pero la mayoría de los ordenadores operan en base a la clase de voltaje obtenible de pilas secas normales: existen muy pocas razones por las cuales no se pueda pensar que casi cualquier ordenador podría convertirse en portátil de esta manera.

El ACT Apricot tipifica una tendencia interesante; se trata de un sistema normal para oficina, si bien de pequeñas dimensiones. Dejando de lado el monitor, el Apricot se convierte en un sistema portátil. Por supuesto, "portabilidad" es un término muy vago. Muchos micros se describen como portátiles cuando simplemente están equipados con un asa para llevarlos, y jamás se construyeron con esa finalidad.

Con la introducción de una amplia gama de micros portátiles, la industria ha dado un gran paso hacia la consecuención de su verdadero potencial: hacer accesible a todos el poder de la informática a un bajo costo y de forma que no se requieran conocimientos o experiencia previos. Hemos de admitir que aún ha de pasar algún tiempo antes de que se cumplan por completo estos criterios; pero ahora el objetivo está más cerca.

Epson HX-20

Este ordenador no sólo se puede llevar a cualquier sitio, sino que le permite al usuario hacer su trabajo de oficina mientras viaja

El Epson HX-20 fue el primer ordenador auténticamente portátil. Como es totalmente programable en BASIC, posee más aplicaciones que las calculadoras más sofisticadas. Aun así, su diseño autocontenido y sus menos de dos kilogramos de peso total significan que se lo puede llevar en el interior de un maletín.

Muchas máquinas de mayores dimensiones que hasta entonces se vendían como "portátiles", hubieron rápidamente de empezar a anunciarse como "transportables". Un amplio sector del mercado de usuario de microordenadores ha comprado el HX-20: aficionados a la informática personal, hombres de negocios e ingenieros.

Debido a que el HX-20 no se basa en un diseño estándar sino que el suyo es nuevo por completo, el software disponible es todavía relativamente escaso. No obstante, es una máquina ideal para aprender a programar en BASIC. Con frecuencia el comprador tiene la intención de escribir un programa para una aplicación específica (y tal vez inusual): un programa de presupuestos para agentes de seguros, un medio auxiliar de navegación para deportistas náuticos y un apuntador de notas para periodistas; todos ellos son programas escritos para el Epson.

La unidad viene con una pantalla de visualización en cristal líquido que puede manejar hasta 4 líneas de 20 caracteres, o gráficos sencillos de una resolución de hasta 120 × 32 puntos. En la mayoría de las aplicaciones esta visualización actúa como una "ventana" que se puede mover mediante las teclas del cursor (señalizadas con flechas) para mostrar cualquier sección de una superficie de texto mucho mayor que maneja el ordenador.

El mecanismo de la impresora incorporada utiliza papel común en un rollo de 5 cm de ancho, sobre el cual puede imprimir hasta 24 columnas de texto.

La unidad de microcassette es un extra opcional, pero normalmente se la muestra como si fuera incorporada debido a que la mayoría de la gente la necesita. El espacio que ocupa se puede utilizar para un software de cartucho transistorizado, aunque por el momento no se ha producido ninguno. La microcassette es superior a la grabadora de cassette corriente. El ordenador detecta la posición de la cinta y se mueve *fast forward* (rápido adelante) automáticamente para hallar el programa o la información requeridos.

La gama de interfaces situada en la parte posterior y en los laterales de la carcasa refleja la diversidad de las aplicaciones de la máquina; incluso se le puede acoplar un lector de código de barras (véase p. 40). Los 16 Kbytes de RAM se pueden ampliar a 32 Kbytes mediante un accesorio que se coloca en un lateral.

Hay a la venta hardware y software nada caros para que el HX-20 se pueda comunicar a través del teléfono con otra máquina similar o con un ordenador principal, y tener acceso a información central.



Chris Stevens

El teclado del Epson HX-20
Es su completo teclado, de dimensiones similares al de una máquina de escribir, lo que en realidad determina el tamaño del HX-20. Si bien el "tacto" de las teclas es algo distinto del de una máquina de escribir convencional, satisface plenamente a los mecanógrafos al tacto.
Además de los numerales situados en la hilera de teclas superior, al pulsar NUM las teclas U, I, O, J, K, L y M se convierten en un relleno de teclado numérico. Esto permite que se pueda efectuar con mayor rapidez la entrada de grandes cantidades de información numérica.
El movimiento del cursor y otras teclas para edición se halla arriba a la derecha, junto con una tecla SCRN (*scrolling*) para hacer girar la pequeña pantalla hacia arriba y hacia abajo.
Las cinco teclas de función programable (de PF1 a PF5) son físicamente diferentes de las otras teclas y la función de dos de ellas es doble, para controlar la microcassette y para copiar en la impresora el contenido de la pantalla





Impresora incorporada
Las pequeñas impresoras matriciales o de rollo de papel han sido parte integrante de las cajas registradoras (e incluso de las calculadoras) durante bastante tiempo. La casa Epson, que dio sus primeros pasos como fábrica de impresoras, incorporó muy rápidamente esa idea a su portátil HX-20. Esta impresora matricial ofrece una salida tanto de gráficos como de caracteres

Enchufe red
Con el transformador conectado, el HX-20 se alimenta desde la red eléctrica y recarga su batería interna

Botón de borrado

Conexión códigos de barras
Aquí se puede conectar un lápiz óptico para leer códigos de barras, que existen actualmente en muchos comercios

Conexión cassette
Trabaja con una grabadora de cassette corriente e incluye el control del motor. No obstante, su rendimiento no es tan eficaz como el de la microcassette

Placa de circuito impreso
En nuestra ilustración, el PCB (*Printed Circuit Board*: placa de circuito impreso) se ha invertido por razones de mayor claridad. Normalmente los circuitos miran hacia la cara inferior del ordenador y las puertas separables en el fondo de la carcasa significan que se pueden cambiar los chips más importantes sin tener que desmontar totalmente la unidad

Microprocesadores
Dos microprocesadores 6301 (fabricados por Epson) controlan el ordenador y sus interfaces. Extraordinariamente, cada uno posee 4 Kbytes de ROM y 128 bytes de RAM incorporados, además de los chips exteriores

RAM
El HX-20 viene con 16 Kbytes de RAM estándar, configurados en forma de ocho chips de 2 Kbytes



Almacenamiento de información
La grabadora de microcassette incorporada del HX-20, construida alrededor de la clase de cintas de cassette que utilizan las máquinas para dictado de bolsillo, representa un avance significativo respecto a las grabadoras de cassette corrientes, porque el ordenador controla la función *fast wind* (bobinado rápido). Por lo tanto se reduce de manera drástica el tiempo necesario para llegar a un determinado punto de la cinta. Otra ventaja es que la unidad se enchufa directamente en un espacio reservado de la carcasa, sin que medie ningún tipo de cable

Disparador de cartucho
Este mando permite retirar entera la microcassette. En teoría este módulo se puede sustituir por un cartucho transistorizado, en el caso de que el usuario desee utilizar la máquina para una sola finalidad

Ajustador de ángulo de visión
Normalmente, para ser legibles, las pantallas de visualización en cristal líquido han de verse desde el ángulo corrector. Haciendo rotar este mando el usuario puede mirar desde cualquier ángulo

Interruptor de potencia

Interface en serie
A pesar de que la conexión RS232 también es una interface en serie, existe esta otra conexión para activar diversos periféricos que la Epson tiene previsto introducir para el HX-20

Conexión RS232
Ésta es la conexión estándar para interfaces en serie y se emplea para acoplar una impresora o un modem, con el fin de comunicarse con otros ordenadores a través del teléfono

Interface para ampliación de memoria
La memoria estándar de 16 K se puede duplicar con la adición de un cartucho Epson, fabricado para armonizar con el resto de la máquina

ROM
Cuatro chips dan un total de 32 Kbytes. Un conector libre para ROM implica que el ordenador se puede convertir en una máquina de aplicación exclusiva si el software se fabrica en forma de ROM

Una nueva técnica
Gran parte del volumen de un microordenador lo constituye el tubo de rayos catódicos que conforma el corazón del monitor. El Epson HX-20 superó este problema sustituyendo la unidad de representación visual común por una pantalla en cristal líquido. Las visualizaciones de este tipo se han venido usando desde hace mucho tiempo en las calculadoras, pero esta aplicación significa un gran paso hacia adelante en términos de calidad, ya que la matriz de 20 × 4 posee un ajustador de ángulo de visión

EPSON HX-20

DIMENSIONES
290 × 210 × 45 mm

PESO
1,7 kg

CPU
6301 Hitachi

VELOCIDAD DEL RELOJ
2,45 MHz

MEMORIA
16 K de RAM, ampliables a 32 K mediante un accesorio..
32 K de ROM, ampliables a 64 K por el mismo método

VISUALIZACION EN VIDEO
Pantalla en cristal líquido (con ajustador de ángulo de visión) con 4 líneas de 20 caracteres que actúan a modo de ventana de una superficie de hasta 255 × 255 caracteres. Cada pixel de la visualización se puede direccionar, obteniendo una resolución para gráficos de hasta 120 × 32 puntos

INTERFACES
Conexiones para ampliación de memoria, cassette, RS232 (para impresora exterior), serial (para futuro disco), lápiz óptico para códigos de barras

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
Intérprete de FORTH

VIENE CON
Maletín de plástico moldeado para transportarlo, fuente de alimentación eléctrica, manual de instrucciones

TECLADO
60 teclas tipo máquina de escribir, cinco teclas de función programable y cuatro botones para función exclusiva

DOCUMENTACION
El ordenador viene con dos manuales: un manual de funcionamiento y un manual de referencia de BASIC. Ambos son exhaustivos, pero no poseen índice. La firma Epson optó por incorporar BASIC Microsoft, estándar en la industria, y el manual para el lenguaje probablemente sea adecuado como instrucción general, y guía de referencia.
El manual de funcionamiento incluye información para especialistas y una completa lista de las conexiones de I/O para el RS232 y las interfaces en serie; los niveles de señalización son aceptables, así como las extensas tablas de los mapas de memoria

Números al azar

Siguiendo con nuestro estudio de las funciones del BASIC llegamos a la función RND, que produce números al azar (o casi al azar) para su utilización en juegos o en programas estadísticos

Ahora que ya hemos visto cómo trabajan varias de las funciones de BASIC, examinaremos una de las que más se utilizan: la función RND. RND se usa para generar números al azar. También se emplea en los juegos cada vez que existe un elemento de casualidad.

Lamentablemente, RND es una de las palabras más "variables" del BASIC. La descripción que nosotros hagamos de ella puede diferir respecto a la función que tenga destinado realizar en algunos micros personales. Aclaremos, entonces, las diferencias existentes entre el BASIC utilizado en nuestro curso de programación BASIC y otros BASIC.

La mayoría de nuestros programas se basan en el BASIC Microsoft (o MBASIC). Microsoft es una empresa norteamericana y su BASIC fue uno de los primeros que estuvieron disponibles a nivel masivo. El BASIC es un lenguaje que no está estandarizado oficialmente, pero el de Microsoft se aproxima, dentro de lo posible, a lo que sería una versión estandarizada. Existen muchas versiones elaboradas a partir de la de Microsoft, y esta empresa tiene encomendado producir otras para varios ordenadores populares.

La diferencia fundamental entre el MBASIC y la mayoría de las versiones más recientes es que ahora los ordenadores personales son capaces de realizar excelentes gráficos, facultad que no poseían cuando se desarrolló el MBASIC. Otras versiones de BASIC incluyen generalmente un número de órdenes y sentencias para gráficos. Para obtener el máximo rendimiento de las prestaciones del ordenador, es necesario aprovechar en profundidad su capacidad para realizar gráficos, y ello requiere un exhaustivo estudio del manual por parte del usuario.

De los diversos BASIC que proporcionan los ordenadores personales más populares, probablemente el BASIC del Sinclair (utilizado en el ZX81 y en el Spectrum) y el BASIC del BBC sean los que más difieren del MBASIC. La versión del Texas Instruments (empleada en el TI99/4A) también incorpora una cantidad de diferencias significativas. En la medida de lo posible, en los recuadros de "Complementos al BASIC" le indicamos cómo debe modificar nuestros programas, y se



debe remitir a estos recuadros en caso de que encuentre alguna dificultad al ejecutar los programas.

Como hemos mencionado antes, la función RND varía de una versión a otra. Compruebe en su manual de BASIC qué cometido cumple en su versión. Nosotros vamos a ilustrar su utilización a partir de un sencillo juego de dados. Al igual que en programas anteriores, hemos realizado la mayor parte del trabajo en forma de subrutinas. Esta técnica ofrece la ventaja de hacer que los programas resulten más fáciles de leer, de escribir y de depurar.

El programa principal comienza con la sentencia RANDOMIZE en la línea 20. La mayoría de las versiones de BASIC, pero no todas, necesitan de esta sentencia para "redistribuir" la función RND. En realidad es bastante difícil lograr que los ordenadores produzcan números auténticamente al azar. Sin esta operación de redistribución, se produciría la misma secuencia de números supuestamente al azar cada vez que se aplicara RND. La línea 50 remite luego a una subrutina que utiliza RND para asignarle a la variable D un número al azar. La forma en que la hemos empleado es:

```
320 LET D = INT (10 * RND)
```

Ésta es la línea que, con toda probabilidad, el usuario habrá de modificar cuando dé entrada al programa. En "Complementos al BASIC" se proporcionan detalles de cómo funcionan las distintas versiones de RND, de manera que veamos qué es lo que está sucediendo en este BASIC Microsoft. La función RND usa una expresión (entre paréntesis, como es normal en estas funciones) como una opción para alterar ligeramente la secuencia de números producida. Sin ninguna expresión (por ejemplo, LET A = RND), el valor de A será un número entre 0 y 1. Nosotros no deseamos un número inferior a 1, por lo cual multiplicamos el número por 10. Esto se puede hacer de la siguiente manera: LET A = 10 * RND. Si, en aras de la demostración, RND hubiera devuelto el valor 0,125455, ahora el valor de A sería 1,25455.

Para eliminar la fracción del número y conservar sólo la parte entera, utilizamos la función INT (entero) de esta manera: LET A = INT (10 * RND). Algunas versiones de BASIC permiten especificar el límite máximo de los números generados al azar en la expresión encerrada entre paréntesis después de RND. Por ejemplo, el BASIC del Dragon imprimirá un número entero entre 1 y 6 en respuesta a PRINT RND(6).

Dado que con nuestro BASIC Microsoft no podemos hacer esto, comprobamos si los números devueltos son mayores que 6 o menores que 1, porque esos números no sirven para un juego de dados. Esto se realiza en las líneas 330 y 340:

```
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
```

Si D quedara fuera de los límites entre 1 y 6, GOTO hará que el programa salte hacia atrás y lo intente otra vez.

Habiendo escogido para D un valor al azar entre 1 y 6, la subrutina de tirar los dados vuelve (RETURN) al programa principal. Éste imprime el mensaje SU NUMERO ES UN, seguido de la imagen de un dado. Observe cómo se selecciona la imagen apropiada de un dado. Esto se efectúa en la subrutina de SELECCION. Por ejemplo, si el dado (y por tanto D) es un 1, la línea 410 llama a la subrutina que comienza en la línea 530, de manera que:

```
410 IF D = 1 THEN GOSUB 530
```

Esta subrutina no es más que una serie de sentencias PRINT diseñada para producir gráficos muy rudimentarios en la pantalla. Bien puede ser que el BASIC del ordenador del usuario posea gráficos de pantalla mucho mejores y, si éste es el caso, lo mejor sería reemplazar nuestras subrutinas por las sentencias para gráficos apropiadas.

Una vez que el programa ha elegido un dado al azar para el usuario, repetirá luego el proceso para escoger y visualizar un dado para el ordenador. La parte del programa que decide quién ha ganado se ha incorporado al programa principal; también se la podría haber escrito en forma de subrutina, pero no habría valido la pena, puesto que sólo tiene cuatro líneas. La línea 200 compara M (mis dados) con C (los del ordenador) para ver si son iguales. Si lo son, a la variable alfanumérica S$ se le asignan las palabras ES UN EMPATE. La línea 210 compara para comprobar si M es mayor que C. Si lo es, le asigna a S$ las palabras HA GANADO USTED. La línea 220 comprueba si M es menor que C. En ese caso, le asigna a S$ las palabras HA GANADO EL ORDENADOR. La línea 240 simplemente imprime el resultado y el juego acaba. Aunque este programa es bastante largo, en esencia es muy sencillo. Utiliza solamente una función, RND, no tiene bucles, ni variables subíndice, ni ninguna otra complicación excepto algunas sentencias IF...THEN.

Dado que la función RND es tan variable y que algunas versiones de BASIC (la Microsoft, por ejemplo) requieren la sentencia RANDOMIZE para generar una nueva secuencia de números al azar, ¿existe algún procedimiento para que podamos generar números auténticamente al azar (es decir, impredecibles) sin usar estas funciones? Sí, existen varias técnicas.

Una de las funciones que hasta ahora no habíamos examinado es la función INKEY$ (que corresponde a *inkey-string*). Cada vez que se encuentra con la palabra INKEY$, el programa inspecciona el teclado para ver si se ha pulsado alguna tecla. El programa no espera a que se dé entrada a un carácter, como lo hace cuando se emplea la orden INPUT. Por tanto la orden INKEY$ normalmente se utiliza dentro de un bucle. El programa entonces explora continuamente el teclado, esperando alguna entrada. Por lo general en el bucle se incluye alguna comparación para que éste termine cuando se dé entrada a un carácter apropiado. Esto permite escribir un programa que forme un bucle para contar que concluirá cuando se haya digitado un carácter específico. ¿Qué sucedería si empleáramos este programa?

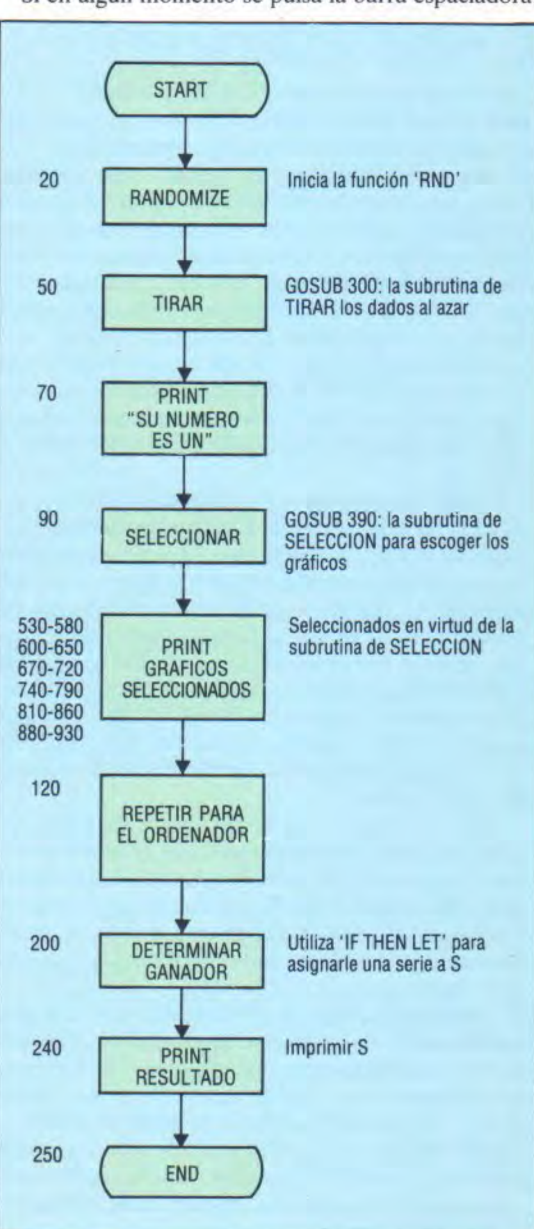
```
10 PRINT "PULSE LA BARRA ESPACIADORA"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 LET A$ = INKEY$
50 IF A$ = "   " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 FOR Q = 0 TO 1
90 IF R < 10 THEN GOTO 130
100 LET Q = 0
110 LET R = R/10
120 NEXT Q
130 PRINT INT (R)
140 END
```

¿Será R un número al azar? Debería serlo, de modo que procedamos a analizar el programa y comprobemos por qué.

La línea 10 imprime la nota PULSE LA BARRA ESPACIADORA. Antes de que podamos responder a esta nota, el programa ha dado entrada el bucle FOR X = 0 TO 1 en la línea 20. Puede que 0 y 1 parezcan límites extraños para el bucle, pero enseguida veremos cómo se utiliza esta estructura. La línea 30 le asigna a la variable R el valor 1 la primera vez que se efectúa el bucle. La línea 40 le asigna a la variable alfanumérica A$ el carácter digitado en el teclado, cualquiera que haya sido. Esto se hace empleando la función INKEY$. Si el usuario pulsara la letra R, a A$ se le asignaría R. La línea 50 compara A$ para ver si es un espacio (que en BASIC se representa como un espacio encerrado entre comillas dobles: "   "). Si A$ es un espacio, el programa se bifurca valiéndose de la sentencia GOTO; pero si A$ no es un espacio, el programa continúa por la siguiente línea.

Se trata de la línea 60, que dice LET X = 0. Ahora X es el índice del bucle. La sentencia NEXT X de la línea 70 hace que el programa retorne al comienzo del bucle, en la línea 20. Dado que X se ha reestablecido en 0, el bucle la repite. De esta manera, el bucle FOR X = 0 TO 1 se repetirá indefinidamente, en la medida en que fracase la comparación IF A$ = "   "

Si en algún momento se pulsa la barra espaciadora,



Flujo del programa
El diagrama de flujo muestra de forma simplificada las acciones principales que realiza el programa. A la izquierda se indican los números de las líneas correspondientes y, a la derecha, se dan unas breves notas explicativas. No se trata de un diagrama de flujo completo, ya que no incluye muchas de las "decisiones" y bifurcaciones del programa

a A$ se le asignará el carácter que represente un espacio, y el programa se bifurcará hasta la línea 80 y el bucle no se repetirá más.

Pero ¿qué sucede mientras el bucle se repite? A cada repetición del bucle, la línea 30 aumenta el valor de R. La primera vez que se realiza, R se establece en 1, la segunda vez se establece en 1 + 1 y así sucesivamente. Cuando el bucle se interrumpe a raíz de la comparación de A$, podemos leer R para ver hasta dónde hemos contado.

Sin embargo, los ordenadores funcionan a gran velocidad, de modo que, para cuando pulsemos la barra espaciadora, R podría haber llegado a varios centenares. ¿Qué haremos si deseamos que los valores de R se hallen sólo entre 1 y 10? La línea 80 introduce otro bucle que nos permite comparar R y dividirla por 10 en el caso de que sea mayor que 10. En la medida en que R sea mayor que 10, fallará la comparación de la línea 90, el valor de Q se reestablecerá en 0 y se repetirá el bucle. La línea 110 divide por 10 el valor de R, pero el resultado no se imprimirá en tanto no se dé la condición de que el valor de R se haya reducido a un número menor que 10. La línea 30 asegura que el valor de R nunca podrá ser 0.

En teoría, entonces, este programa debería producir un número al azar entre 1 y 9, ambos inclusive. Pero ¿lo hace? La sentencia INT asegura que se han eliminado las fracciones decimales; por tanto, los posibles valores de R serían 1, 2, 3, 4, 5, 6, 7, 8, 9. El promedio de estos números es 5 (porque suman 45, y $45 \div 9 = 5$). Pruebe y vea lo que sucede. Lo puede hacer ejecutando el programa una cantidad de veces, anotando el valor de R obtenido cada vez y calculando luego el promedio. Como alternativa, puede agregarle algunas líneas al programa para ejecutarlo, pongamos por caso, 100 veces, sumando a otra variable S el valor de R y dividiendo luego S por 100.

Cuando realizamos esto, nos encontramos con que el valor promedio de R es bastante inferior a 5 y que, por lo tanto, los números no pueden haber sido producto del azar. Es interesante analizar las causas de ello.

El problema es que, si bien el BASIC es rápido, no lo es en la medida necesaria. El primer bucle deja que el valor de R aumente hasta que llega a varios cientos, incluso miles, antes de que nosotros pulsemos la barra espaciadora. A menos que se realice un esfuerzo deliberado para modificar el lapso transcurrido entre el momento en que el usuario ve la nota PULSE LA BARRA ESPACIADORA y en efecto la pulsa, lo cierto es que se la habrá pulsado después de haber pasado demasiado tiempo, de modo que en el curso del lapso transcurrido lo más probable es que el valor de R haya aumentado varios cientos.

Las divisiones que se llevan a cabo para reducir el valor de R a un número menor que 10 no se realizan hasta después de que se haya pulsado la barra espaciadora. Esto significa que R casi siempre está en las centenas bajas antes de que se efectúe la división, de manera que, en consecuencia, el valor final de R tenderá a ser bajo.

¿Se puede escribir una rutina para superar este inconveniente? La respuesta es afirmativa, siempre y cuando podamos lograr que el proceso de contar sea lo suficientemente rápido como para que nuestro tiempo de reacción ante la nota PULSE LA BARRA ESPACIADORA sea, en realidad impredecible. La solución consiste en hacer la condición para la parte "mayor que el límite máximo" del primer bucle. Consideremos este programa:

```
10 REM JUEGO DE DADOS -- PROGRAMA PRINCIPAL
20 RANDOMIZE
30 REM SU TURNO
40 REM GOSUB RUTINA DE'TIRAR'
50 GOSUB 300
60 LET M = D
70 PRINT "SU NUMERO ES UN"
80 REM GOSUB RUTINA DE'SELECCION'
90 GOSUB 390
100 PRINT
110 REM TURNO DEL ORDENADOR
120 REM GOSUB RUTINA DE'TIRAR'
130 GOSUB 300
140 LET C = D
150 PRINT "EL NUMERO DEL ORDENADOR ES UN"
160 REM GOSUB RUTINA DE'SELECCION'
170 GOSUB 390
180 PRINT
190 REM  ¿QUIEN GANO?
200 IF M = C THEN LET S$ = "EMPATE"
210 IF M > C THEN LET S$ = "GANO USTED"
220 IF M < C THEN LET S$ = "GANO EL ORDENADOR"
230 REM PRINT RESULTADO
240 PRINT S$
250 END
260 REM
270 REM
280 REM
290 REM
300 REM SUBRUTINA DE TIRAR LOS DADOS AL AZAR
310 REM
320 LET D = INT(10* RND)
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
350 RETURN
360 REM
370 REM
380 REM
390 REM SUBRUTINA DE SELECCION
400 REM
410 IF D = 1 THEN GOSUB 530
420 IF D = 2 THEN GOSUB 600
430 IF D = 3 THEN GOSUB 670
440 IF D = 4 THEN GOSUB 740
450 IF D = 5 THEN GOSUB 810
460 IF D = 6 THEN GOSUB 880
470 RETURN
480 REM
490 REM
500 REM
510 SUBRUTINAS DE 'GRAFICOS'
520 REM
530 PRINT "            "
540 PRINT "            "
550 PRINT "            "
560 PRINT "     *      "
570 PRINT "            "
580 PRINT "            "
590 RETURN
600 PRINT "            "
610 PRINT "            "
620 PRINT "         *  "
630 PRINT "            "
640 PRINT "  *         "
650 PRINT "            "
660 RETURN
670 PRINT "            "
680 PRINT "            "
690 PRINT "         *  "
700 PRINT "     *      "
710 PRINT "  *         "
720 PRINT "            "
730 RETURN
740 PRINT "            "
750 PRINT "            "
760 PRINT "  *      *  "
770 PRINT "            "
780 PRINT "  *      *  "
790 PRINT "            "
800 RETURN
810 PRINT "            "
820 PRINT "            "
830 PRINT "  *      *  "
840 PRINT "     *      "
850 PRINT "  *      *  "
860 PRINT "            "
870 RETURN
880 PRINT "            "
890 PRINT "            "
900 PRINT "  *      *  "
910 PRINT "  *      *  "
920 PRINT "  *      *  "
930 PRINT "            "
940 RETURN
```

```
10 PRINT "PULSE LA BARRA ESPACIADORA"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = " " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 PRINT R
90 END
```

En este programa, R nunca puede ser menor que 1 ni mayor que 9. Para cuando se pulse la barra espaciadora (y lo reconozca la función INKEY$ de la línea 50), R tendrá un valor cualquiera comprendido entre 1 y 9, ambos inclusive.

Este programa fue comprobado mil veces y se halló para R un valor promedio de 5,014. Puesto que el promedio perfecto sería 5 y que el margen de error es sólo un 0,28 % superior, esto sugeriría que de hecho el programa genera un número al azar que se aproxima bastante al promedio teórico. La cuestión estriba, por supuesto, en que incluso cuando un programa parece razonable sobre el papel, luego, en la práctica, puede tener algunos fallos imprevistos. Este motivo justifica el que siempre valga la pena realizar una comprobación a conciencia.

Algunos lectores habrán observado que estos programas para números al azar se pueden acortar mediante la utilización de varias sentencias GOTO en lugar del bucle FOR...NEXT. En próximos capítulos de nuestro curso de programación BASIC explicaremos las razones por las cuales hemos preferido evitar las sentencias GOTO.

Complementos al BASIC



En el micro BBC y el Oric-1, elimine la línea 20 y sustituya la 320 por:

320 LET D = INT (10 * RND(1))

En el Dragon-32, suprima la línea 20 y sustituya la 320 por:

320 LET D = RND(6)

y elimine las líneas 330 y 340.

RND

En el Lynx, sustituya la línea 20 por:

20 RANDOM

En el Vic-20 y el Commodore 64, reemplace la línea 20 por:

20 LET X = RND(−TI)

y sustituya la línea 320 por:

320 LET D = INT(10 * RND(1))

En el Spectrum, la palabra RANDOMIZE aparece abreviada en el teclado (RAND), pero en la pantalla se leerá RANDOMIZE



En el Oric-1 y en el Lynx, reemplace INKEY$ por KEY$.
En el Vic-20 y el Commodore 64, sustituya la línea 40 por:

40 GET A$

Luego reemplace la línea 50 por:

50 GET A$: IF A$ = " " THEN GOTO 80

En el micro BBC, sustituya INKEY$ por INKEY$(10). El número entre paréntesis indica el tiempo, expresado en centésimas de segundo, durante el cual el sistema aguardará a que se pulse una tecla; así que para obtener una respuesta rápida deberá usar un número bajo, y viceversa

Ejercicios

■ **Función RND** Modifique el último programa del texto para que dé un número al azar entre 1 y 6 (ambos inclusive).

■ **Bucle y promedio** Agregue algunas líneas al último programa del texto con la finalidad de hacer que se repita cien veces y a continuación dé el promedio de los cien resultados.

■ **Sustitución por una subrutina** Sustituya las líneas 50 y 130 del programa principal (es decir, la subrutina de tirar los dados al azar) por un GOSUB que invoque a su "generador de números al azar" existente en el primer ejercicio.

■ **INKEY$** Utilizando la función INKEY$, ¿cómo escribiría usted un programa que leyera cualquier tecla digitada en el teclado e imprimiera: LA TECLA QUE USTED HA PULSADO ES: * (* representa la tecla pulsada por usted).

■ **Bucle de tiempos** Escriba un bucle de tiempos (un bucle "para contar") y utilice la función INKEY$ para hallar cuál es el valor más alto que alcanza una variable al cabo de 10 segundos (necesitará un reloj). Escriba el programa de modo que la última salida impresa diga: EL VALOR DE R AL CABO DE 10 SEGUNDOS ES: * (* representa el valor de R).

■ **Comparaciones IF-THEN** Escriba un programa para juegos sencillo en el cual el ordenador genere un número al azar entre 1 y 100 (ambos inclusive) y el jugador tenga que adivinar de qué número se trata. El jugador dispone de tres oportunidades. Cada vez, el programa le responde con los mensajes SU ELECCION ES DEMASIADO ELEVADA, SU ELECCION ES DEMASIADO BAJA, o HA ACERTADO, FELICITACIONES, o YA NO TIENE MAS OPORTUNIDADES. ¡HA PERDIDO!

Las respuestas, en el próximo capítulo.

Respuestas a los "Ejercicios" de las páginas 148 y 149

Bucles 1

EL VALOR DE A ES 450

Bucles 2

START
STOP

Bucles 3

AHORA EL VALOR DE A ES 160

Bucles 4

ME GUSTA EL BASIC
ME GUSTA EL BASIC
ME GUSTA EL BASIC
:
:

Hasta que borre (RESET) o interrumpa (BREAK) el programa

Bucles 5

ESTOY ALGO TONTO
(15 veces)

Read-Data 1

ESTAMOS COMPARANDO LA SENTENCIA 'READ'
170

Read-Data 2

X = 1
X = 2
:
:
X = 23

Sobre dos ruedas

La robótica se está desarrollando rápidamente. Los robots móviles, como el Buggy BBC, se pueden programar incluso para que detecten obstáculos

Los robots y las tortugas móviles no tienen sólo una función educativa sino que también son una fuente de diversión. Los principios que se aplican para controlar un dispositivo como el Buggy de la BBC son los mismos que rigen para los robots industriales de grandes dimensiones. Aunque en realidad todavía no sirven para realizar los trabajos domésticos, podrían conventirse en la próxima generación en dispositivos auxiliares para el hogar.

Es necesario que los robots se sitúen con exactitud en relación a su entorno, y por esta causa normalmente se accionan mediante motores paso a paso. A diferencia de los motores convencionales, los motores paso a paso no rotan cuando se les suministra energía. A cada impulso de energía que se les aplica, el eje gira una fracción preestablecida de rotación completa. El número de impulsos que se requieren para efectuar una revolución completa depende de cada motor. También se puede controlar el sentido de la rotación. Es posible lograr que un robot o una tortuga móviles se desplacen a través de distancias muy precisas, en cualquier dirección, permitiendo que el ordenador accione los motores por separado. Un robot móvil puede girar en el sitio en que se halla accionando sólo las dos ruedas en sentido contrario.

Sin embargo, resulta igualmente importante que el robot se pueda remitir al ordenador cuando se encuentra con algo. Las colisiones se suelen detectar montando unos parachoques alrededor del cuerpo del vehículo y conectándolos a microinterruptores. Éstos, a su vez, están acoplados a la conexión de entrada del ordenador y la apertura o el cierre de cada interruptor hará que uno de los bits cambie a "0" o a "1".

Con frecuencia se requieren otros tipos de entradas desde el robot. Podría ser útil, pongamos por caso, la capacidad para seguir líneas blancas trazadas sobre un suelo oscuro. Esto se consigue haciendo que disminuya la intensidad de una fuente luminosa situada en el robot, que posee junto a ella un detector para medir la cantidad de luz reflejada. Esta cantidad variará a tenor de la superficie por la cual se desplace el robot a cada momento; se trata más bien de una cantidad analógica antes que de una cantidad digital. El microordenador BBC está equipado con una entrada analógica que permite utilizar directamente un detector de este tipo. En la mayoría de los otros sistemas la señal se ha de convertir en una señal digital para que se la pueda volver a enviar al ordenador.

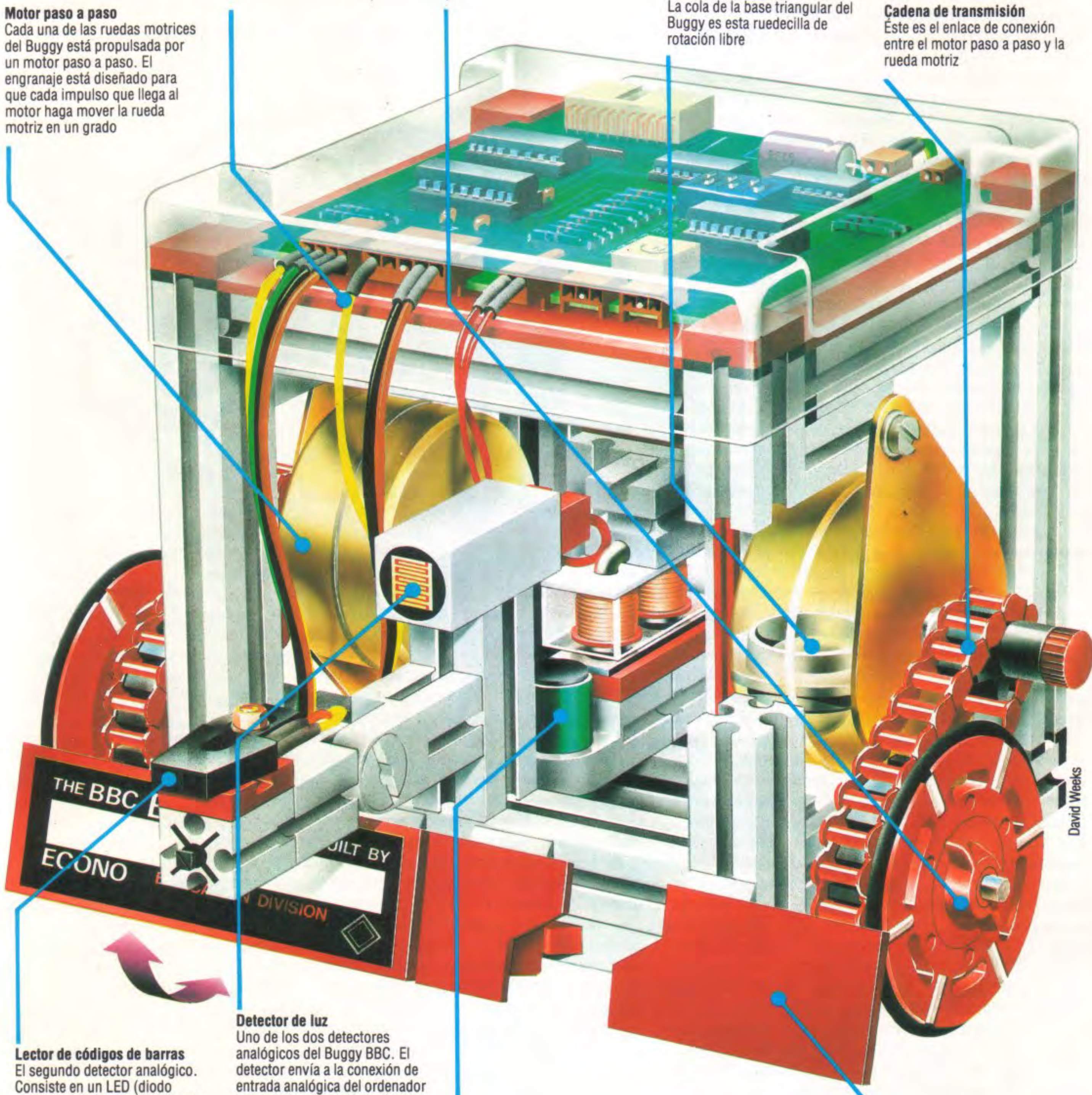
Otra aplicación para este tipo de detectores es el lector de códigos de barras. Los códigos relativos a la naturaleza de los productos almacenados en un depósito se podrían explorar del mismo modo que el robot busca el ítem correcto. El Buggy BBC viene con un software de demostración que emplea su lector de có-



En marcha

Esta "microtortuga" está en realidad a medio camino entre un robot móvil y una tortuga especializada, debido a que posee detectores de colisión y, en consecuencia, se le puede otorgar un nivel de "inteligencia". El Zeaker se puede desplazar hacia atrás, hacia adelante, hacia la izquierda y hacia la derecha, y también puede mover un lápiz hacia arriba y hacia abajo. Su control es muy sencillo gracias a una versión del lenguaje LOGO denominada "SNAIL LOGO", que permite dar directamente órdenes como FORWARD (hacia adelante) y BACKWARD (hacia atrás). A excepción del microordenador BBC, todos los otros ordenadores requerirán una unidad para interface. Esto representa un gasto extra, pero esta unidad posee su propia fuente de alimentación y, por tanto, no supone ninguna exigencia adicional para la fuente de alimentación eléctrica del ordenador

Chris Stevens



Motor paso a paso
Cada una de las ruedas motrices del Buggy está propulsada por un motor paso a paso. El engranaje está diseñado para que cada impulso que llega al motor haga mover la rueda motriz en un grado

Tablero de control
Todas las señales que llegan y salen del ordenador se procesan mediante esta ficha para interface antes de ser repetidas

Rueda motriz
Las ruedas con cubiertas de goma situadas a cada lado del Buggy proporcionan el movimiento y evitan que el dispositivo patine, permitiendo que los desplazamientos sean precisos

Ruedecilla trasera
La cola de la base triangular del Buggy es esta ruedecilla de rotación libre

Cadena de transmisión
Éste es el enlace de conexión entre el motor paso a paso y la rueda motriz

Lector de códigos de barras
El segundo detector analógico. Consiste en un LED (diodo emisor de luz) y un LDD (diodo detector de luz). La luz del LED se refleja desde la superficie por debajo del Buggy y es detectada por el LDD. La cantidad de luz reflejada depende del material y de su color, de manera que además de leer códigos de barras el detector también se puede emplear para seguir líneas trazadas en el suelo

Detector de luz
Uno de los dos detectores analógicos del Buggy BBC. El detector envía a la conexión de entrada analógica del ordenador una señal que es proporcional a la cantidad de luz detectada: cuanta más luz brilla en el Buggy, mayor es la señal

Portalápiz
El Buggy BBC se puede equipar con un portalápiz opcional entre las ruedas para que pueda trazar líneas por las superficies por donde pasa. El movimiento hacia arriba y hacia abajo del portalápiz se controla mediante el ordenador

Detectores frontales de colisión
Cada uno de los lados del parachoques delantero dividido está equipado con un microinterruptor que le proporciona al ordenador una señal de encendido (colisión) o apagado (no colisión)

digos de barras para ejecutar música; los principios son los mismos.

Otras clases de señales analógicas que le podrían resultar interesantes de seguir al robot móvil son luz, sonido y campos magnéticos. Estos campos se utilizan con frecuencia para las aplicaciones industriales en las cuales un robot ha de seguir un camino fijo a través de un almacén de depósito o de una fábrica. Los cables especiales enterrados bajo el suelo trazan el "camino" magnético que debe seguir.

El control del robot móvil por lo general se maneja mediante una serie de rutinas de un programa escrito especialmente con esa finalidad. Estas rutinas manejan el envío y la recepción de información a través del cable que conecta el dispositivo con el ordenador. En el caso del Buggy BBC, para controlar los motores se utilizan cuatro bits. Desde el Buggy los datos se envían a través del mismo cable. Las salidas analógicas provenientes tanto del sensor de luz como del lector de códigos de barras van hacia la conexión analógica, y los dos detectores de colisión están conectados a otras dos líneas de entrada al ordenador.

Tanto una conexión de I/O como la conexión dispositivos se pueden examinar mirando determinada dirección del mapa de memoria del ordenador. Generalmente para leer el contenido normal de la dirección se utiliza la orden en BASIC PEEK. Para alterar el contenido, por ejemplo para cambiar el sentido de uno de los motores y hacer girar el robot, el programador debe alterar el valor del bit adecuado en la dirección. En un programa en BASIC esto se conseguirá mediante la orden POKE.

La información analógica se puede examinar de forma bastante parecida, siempre y cuando el ordenador que se esté empleando posea un convertidor analógico-digital incorporado. Si no se dispone de tal dispositivo, se le ha de agregar al robot una unidad para interface que transforme las señales analógicas en información digital antes de reenviarlas al ordenador.

Una tortuga es, en realidad, una forma de robot móvil concebida para utilizarla en conjunción con el lenguaje LOGO, si bien la distinción entre un robot y una tortuga se está volviendo bastante confusa (véase la p. 164). Muchos de los modelos de tortuga más recientes están equipados con detectores de colisión, así como muchos robots móviles se están equipando con portalápices para que puedan funcionar como tortugas. La idea de que pueda ser posible crear dibujos a gran escala haciendo mover un lápiz con ruedas sobre la superficie surge de la enseñanza de la relación existente entre distancia, ángulo y forma. Si una persona se desplaza 10 unidades hacia adelante, gira a la izquierda, se mueve 10 unidades, gira a la izquierda, se mueve 10 unidades, gira a la izquierda y se desplaza otras 10 unidades, habrá caminado describiendo un cuadrado. Para ilustrar estas formas y sus relaciones con el movimiento, podemos acoplarle un lápiz a un robot o a una tortuga y crear la forma sobre papel.

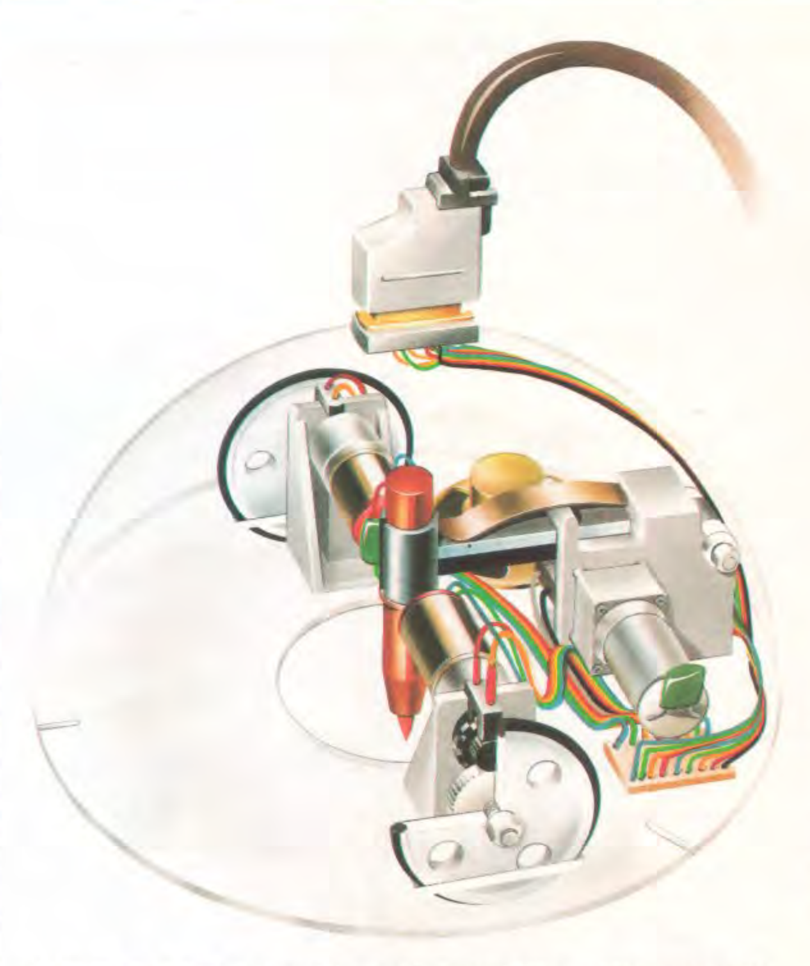
El Buggy BBC se vende en forma de juego de construcción, de modo que el usuario ha de montarlo antes de que pueda explorar el mundo de la robótica. El Buggy está basado en un sistema disponible a nivel comercial, el Fischer Technik, por lo cual ampliarlo y mejorarlo es muy sencillo.

Montar un dispositivo como el Buggy es, en sí mismo, una actividad educativa. Se puede aprender mucho examinando la forma en que se acoplan entre sí los diversos componentes. No obstante, el verdadero aprendizaje empieza cuando el usuario intenta asumir el control de su nuevo "juguete". Aunque muchos de los robots móviles disponibles en el comercio vienen con un software de control, resulta mucho más divertido escribir uno mismo los programas. Para ello se requiere enfocar la programación desde una perspectiva completamente nueva: la del control.

Mientras el robot está funcionando, el programa supervisa sus sensores constantemente para ver si ha hallado una línea en el suelo, si ha detectado una intensa fuente de luz, si ha chocado contra una silla, etc. En el instante en que el detector señala algo, el ordenador ha de reaccionar para proteger al robot de un daño potencial. Estos programas se denominan "en tiempo real" porque sus respuestas han de ser inmediatas.

En teoría existe muy poca diferencia entre un programa que permita que un robot móvil se desplace por una habitación sin chocar contra ningún objeto, y uno que controle una central eléctrica. Las técnicas que se aprenden jugando con dispositivos como el Buggy BBC también pueden desarrollar la comprensión de la inteligencia artificial. Se pueden escribir programas que permitan que un robot móvil realice una tarea predeterminada hasta que un detector perciba que se le están agotando las baterías. El robot busca entonces una fuente de alimentación eléctrica adecuada para recargarse a sí mismo y poder seguir funcionando.

La próxima generación de robots móviles ofrecerá configuraciones aún más notables. Probablemente estarán equipados con brazos en forma de garfios para que puedan coger y transportar cargas pequeñas. Asimismo, quizá los sensores luminosos se reemplacen por cámaras transistorizadas en miniatura gracias a las cuales el robot pueda "ver" a dónde se dirige. Las unidades para síntesis de voz le permitirán comunicarse con su operador, y el reconocimiento de voz abrirá otro canal de control sobre las acciones del robot.



David Weeks

Robot especializado
Una tortuga es como un robot especializado que dibuja en el suelo con un lápiz retráctil de punta de fieltro siguiendo las instrucciones del ordenador. Las tortugas normalmente se asocian con el lenguaje educativo LOGO (véase p. 164), si bien se pueden accionar por BASIC

"Dulces dieciséis"

Los ordenadores trabajan en notación binaria, dado que pueden hacer frente a sólo dos situaciones. ¿Por qué, entonces, algunas veces los programadores utilizan el sistema hexadecimal, que es de base 16?

Resulta sencillo comprender por qué los ordenadores emplean internamente el sistema binario: la representación de los números utilizando sólo ceros y unos se presta bien para las señales eléctricas encedido/apagado del ordenador. Asimismo, resulta sencillo comprender por qué la sociedad humana utiliza casi universalmente el sistema decimal: como tenemos diez dedos, este número se ha convertido en nuestro número de base natural.

Pero ¿para qué habría de servir el sistema hexadecimal (de base 16)? La razón es que convertir números hexadecimales en números binarios y viceversa es mucho más fácil que convertir decimales en binarios y binarios en decimales. Puesto que los ordenadores "comprenden" los números binarios, ¿por qué los informáticos no programan con dígitos binarios? Por dos razones. La primera de ellas es que los números binarios son mucho más largos que sus equivalentes de base 10 o de base 16. Por ejemplo, el número 356 en decimal es 0000000101100100 en binario. La segunda razón es que utilizando números que sólo consten de ceros y unos resulta mucho más fácil equivocarse.

El sistema hexadecimal utiliza 16 dígitos diferentes, incluyendo el cero, de manera que el número mayor que se puede representar en la columna de las "unidades" es el equivalente al decimal 15. Los matemáticos podrían haber inventado seis nuevos símbolos para los números entre 10 y 15, pero la convención exige emplear las seis primeras letras del alfabeto para los seis dígitos extra. Cuando contamos mediante la adición de unos, usamos los dígitos numéricos estándar junto con los dígitos alfabéticos hasta llegar a la F, el equivalente hexadecimal de 15. Después de ello, se nos terminan los símbolos y hemos de comenzar una nueva columna, la de los "dieciséis". Aquí también hemos de utilizar los símbolos disponibles hasta la F. Cuando tanto las columnas de los "dieciséis" como la de las "unidades" estén completas, sumar uno implicará que tengamos que agregar una tercera columna, la de los "doscientos cincuenta y seis". En el panel vemos algunos números decimales con sus equivalentes en binario y en hexadecimal.

En hexadecimal, el número 65 535 se representa FFFF, con lo cual resulta evidente una de las ventajas que ofrece este sistema. Un número cuya representación en decimal exige 5 dígitos y 16 en binario, en notación hexadecimal sólo requiere 4 dígitos. Aparte de esta compactibilidad, ofrece otra ventaja más importante. Cuatro dígitos binarios se pueden representar exactamente con un dígito hexadecimal. Esto hace que la conversión de binario a hexadecimal y viceversa sea relativamente directa.

Para transformar un número binario en su equivalente hexadecimal, divida el número binario en grupos de cuatro dígitos, empezando por la derecha, y convierta a hexadecimal un grupo cada vez. A continuación le ofrecemos algunos ejemplos:

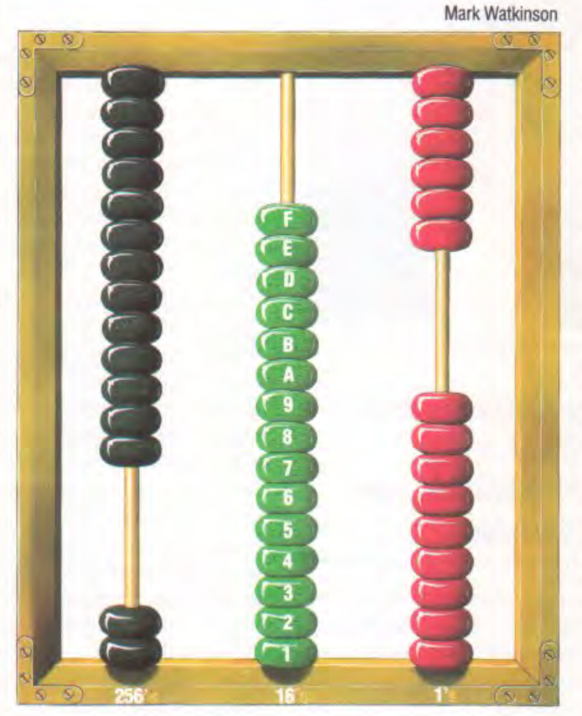
		1110	1001	binario
		=E	9	hex.
		=233		decimal
	1111	1000	1100	binario
	=F	8	C	hex.
	=3980			decimal
0111	1110	0010	1101	binario
=7	E	2	D	hex.
=32 301				decimal

¿Cuándo deseará realmente emplear la notación hexadecimal? A pesar de que los lenguajes de alto nivel, como el BASIC, no le exigen la utilización de hexadecimales (ni, para el caso, de binarios), sí lo exigen lenguajes de bajo nivel como el ensamblador y el código de lenguaje máquina.

Los números escritos en hexadecimal se suelen escribir seguidos de una H para diferenciarlos de los números decimales. Por lo tanto, 256 (en decimal) se escribiría 100H y se leería "uno cero cero hex.".

En esta obra utilizaremos muy poco los números hexadecimales. No obstante, puede que el usuario se encuentre con este término en los apéndices del manual de instrucciones del ordenador. Por eso conviene saber calcular su equivalente en decimal o en binario.

Mark Watkinson



El ábaco hexadecimal
Un ábaco construido para el sistema hexadecimal tendría 15 bolitas en cada varilla. En la varilla situada a la derecha, las bolitas se pueden hacer bajar de una en una hasta llegar a la última, la F. El hexadecimal es un sistema numérico de base 16 y para los números mayores que 9 se utilizan las letras del alfabeto. Como muestra la varilla central, A corresponde al decimal 10, B al 11, C al 12, D al 13, E al 14 y F al 15. Después del "número" F se nos terminan los dígitos y hemos de remitirnos a la columna siguiente, la de los "dieciséis". En la ilustración, la columna de los "1" nos muestra una cuenta de 9 bolitas. La columna de los "16" muestra una cuenta de F y la columna de los "256" una cuenta de 2. Esto se lee "2 F 9 hex.". Es el equivalente al número decimal 761 (256 × 2 + 16 × 15 + 1 × 9)

Binario	Decimal	Hex.
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15

Chuck Peddle



Es el creador del microprocesador 6502, ahora fabricado al por mayor y utilizado por la mayoría de los ordenadores personales

Perteneciente a una generación anterior a la de los "jóvenes genios" empresarios como Steve Wozniak y Steve Jobs, el primer contacto de Chuck Peddle con los microprocesadores se produjo en 1973, cuando ingresó en la Motorola para trabajar en el proyecto de diseño del microprocesador 6800.

Puesto que se trataba del primer microprocesador que salía al mercado británico, Motorola pudo asignarle al 6800 un precio elevado: 2000 libras (unas 45 000 pesetas). Peddle consideró que el producto había sido sobrevalorado excesivamente y dejó la Motorola para integrarse a MOS Technology.

Se unió a esta relativamente pequeña compañía para trabajar en el proyecto de diseño de otro microprocesador, que se convertiría en el 6502 MPU, sin ninguna duda el microprocesador más exitoso de la primera década de la microinformática. No obstante, en aquel momento nadie comprendió que el producto en el cual estaban trabajando estaba llamado a convertirse en el baluarte de toda una industria y que contribuiría en gran medida al estallido de una revolución social como no se había visto desde hace dos siglos.

Una de las personas que comprendieron la trascendencia del microprocesador en general, y el potencial del MOS Technology 6502 en particular, fue Jack Tramiel, presidente de la Commodore. Hasta entonces, Commodore Business Machines se había dedicado a una gama de productos para oficina y calculadoras de bolsillo, con una aceptación discreta.

La Commodore era el principal cliente de MOS Technology, a la que compraba regularmente grandes cantidades de chips especializados para calculadoras de cuatro funciones. Tramiel, a pesar de las dificultades que él mismo tenía para mantener a flote la Commodore, tuvo suficiente fe en el 6502 como para conseguir de cualquier parte el capital necesario para adquirir MOS Technology. Simultáneamente con esta operación, contrató los servicios de Charles Peddle, que era a la sazón ingeniero de desarrollo de microprocesadores.

Ya en aquel entonces Peddle había comprendido que el producto que había desarrollado, fruto de sus propias ideas, podía tal vez marcar un hito: el ordenador personal. Era la misma idea que, de forma independiente, estaban alentando Wozniak y Jobs en la Apple Computer. Peddle estaba tan preocupado con el hecho de que la nueva tecnología se utilizara de manera apropiada, que se asoció con Bill Gates, fundador de Microsoft (famosa por su intérprete de BASIC), con la intención de comprar Apple, que, casualmente, había sido puesta en venta en el mismo momento que MOS Technology. Sin embargo, Wozniak y Jobs pedían 150 000 dólares por la empresa, y la oferta de Peddle y Gates sólo llegaba a los dos tercios de esa cifra.

Peddle permaneció en la Commodore y asumió la tarea de producir el Commodore PET (Personal Electronics Transactor), que apareció en 1977 en el mercado, casi en la misma época en que se lanzó el Apple II. El PET era distinto, en el sentido de que llevaba un monitor y un paquete de cassettes incorporado, y que el "tacto" del teclado se parecía más al de una calculadora que al de una máquina de escribir. Al poco tiempo de haber aparecido, Commodore tenía pedidos en firme para un millar de unidades, y, gracias a la labor desarrollada por Chuck Peddle, había nacido la primera generación de ordenadores diseñados específicamente para utilizarse en el hogar.

Transcurrieron tres años hasta que Peddle pudo concretar su segunda gran ambición: la de dirigir su propia empresa de ordenadores. Con Chris Fish, uno de los cerebros financieros que hicieron factible el súbito crecimiento de la Commodore, se asoció a Victor United, subsidiaria de la gigantesca Walter Kidde Corporation, y puso en marcha Sirius Systems Technology.

El trabajo de desarrollo en el campo de la industria de ordenadores personales se concentró de manera preferente en los chips de 16 bits, como el 8088 de Intel. Resultó que IBM también estaba trabajando en un ordenador personal para escritorio basado en el mismo chip, pero casualmente Sirius consiguió presentar el fruto de su trabajo algunas semanas antes. La máquina tuvo una amplia aceptación y pronto se afianzó en el mercado, siendo el primer microordenador barato y fabricado al por mayor que ofrecía las ventajas de la nueva generación de microprocesadores de 16 bits.

El Sirius 1 era relativamente barato y fácil de usar. Con su teclado separable, sus gráficos de alta resolución y su pantalla antideslumbrante, estableció nuevos niveles para los microsistemas de oficina. Los usuarios descubrieron la inmensa utilidad de las muy mejoradas velocidad y capacidad de direccionamiento del microprocesador de 16 bits.

En conjunto, Chuck Peddle ha recorrido un largo camino con el objetivo de hacer realidad su gran ambición: poner al alcance de todos el potencial del ordenador. Y en el proceso ha abierto nuevos caminos para que otros continúen su trabajo.



Cortesía de Sirius Computing

El primer ordenador personal
Después de desarrollar el microprocesador 6502, Chuck Peddle concentró sus esfuerzos en diseñar lo que se denominó "ordenador personal", una máquina autoconcentrada, que se podía enchufar a un tomacorrientes y utilizar de inmediato para cualquier finalidad que le interesara al usuario. El fruto de su trabajo fue el Commodore PET, que apareció casi simultáneamente que el Apple II de Steve Wozniak y llevaba pantalla, grabadora de cassette e intérprete de BASIC Microsoft incorporados. Aunque desde entonces ha sido rediseñado varias veces, el PET goza aún de gran popularidad. Una de las características más atractivas de la máquina original era su diseño. En la actualidad el 6502 es el microprocesador más utilizado para ordenadores personales

Dibujos animados

Tome los gráficos de su ordenador, multiplique su calidad por mil, y obtendrá un sistema de animación por ordenador

Todo el proceso de realización de filmes, ya sea para el cine o la televisión, se basa en la incapacidad del cerebro de "congelar" una imagen. Una rápida sucesión de imágenes produce la impresión de movimiento.

Los primeros intentos de crear la ilusión de una imagen en movimiento se lograron mediante un cilindro ranurado, en cuyo interior se había pegado una tira de dibujos, que se hacía girar rápidamente. Al mirar a través de las ranuras, se ve una tosca representación de una imagen (o "fotograma") tras otra. El *zoótropo*, nombre que recibió este dispositivo, es un precursor de la ciencia de la fotografía, aunque, naturalmente, al poco tiempo los dibujos del interior del tambor fueran sustituidos por fotografías. El siguiente paso, el soporte fílmico, requería emulsiones fotográficas de acción relativamente rápida, capaces de grabar una imagen en menos de un dieciseisavo de segundo, puesto que los primeros filmes se proyectaban a 16 imágenes por segundo.

Técnicas de simulación del movimiento

Extrañamente, algún tiempo antes, la industria cinematográfica concibió la idea de dibujar a mano cada fotograma, fotografiar los dibujos y proyectar el resultado para producir dibujos animados. Teniendo en cuenta que cada segundo de visión requiere la creación de 24 dibujos diferentes (la velocidad de proyección de una película moderna), está claro que la producción de un filme de tan sólo cinco minutos necesita una cantidad de trabajo enorme, 7 200 fotogramas en este caso. No es sorprendente, por tanto, que en el diseño de las ilustraciones se deban cumplir ciertas reglas: el requisito más importante es que se repitan con exactitud. ¡Sería inadmisible, por ejemplo, que en una película Bugs Bunny cambiara de apariencia a cada segundo!

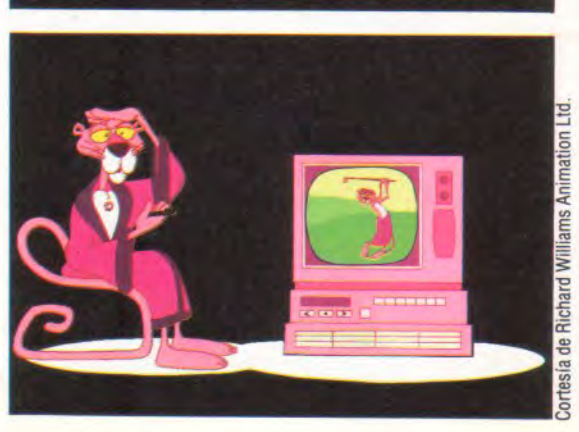
Tareas repetitivas y precisas similares a ésta son realizadas fácilmente por máquinas. Cuando el ordenador asume el control del trabajo de animación —ajustando la velocidad de movimiento, cambiando la perspectiva y geometría, aumentando la luz o la sombra, produciendo cambios en el volumen y el ritmo—, el artista puede concentrarse en la calidad de la imagen. De esta forma, la animación pasa a ser una verdadera arte gráfica, en la cual el artista emplea su tiempo en la creación de la imagen, que el ordenador hará que se mueva.

En su forma más simple, este proceso utiliza gráficos *sprite* (véase p. 152) para crear el "reparto"; los personajes son trasladados a la pantalla y puestos en movimiento, produciendo el tipo de animación empleada en los juegos de video sencillos. Para crear tanto la ilusión de cambio como la de movimiento (por ejemplo, alguien andando), es necesario sustituir repetidamente un sprite por otro. Como se vio, la creación de sprites es una tarea relativamente lenta si consideramos que la calidad gráfica del resultado y la imagen dependen sólo de una representación bidimensional muy sencilla.

La siguiente etapa de la animación requiere que el dibujante-programador construya un algoritmo que introduzca una sensación de profundidad en la imagen según las normas de la perspectiva. Así los objetos pueden definirse en la pantalla según sus coordenadas X, Y y Z. En este punto, es extremadamente práctico







Cortesía de Richard Williams Animation Ltd.

Fotograma a fotograma
Los dibujos animados convencionales, como estos fotogramas de la Pantera Rosa, exigen que el artista dibuje cada figura por separado, aunque no es necesario volver a dibujar los rasgos comunes, a menos que cambien su aspecto o posición. Se utiliza un soporte transparente, de forma que la imagen completa se obtiene a partir de una serie de superposiciones. El artista concentrará su atención en los fotogramas clave de la secuencia, dejando el acabado de las secciones menos importantes a cargo del equipo de ayudantes. Los dibujos acabados se fotografían con una truca, en el orden en que serán vistos

el que el programa no reproduzca líneas "ocultas" e introduzca opacidad en lo que ha sido, hasta ahora, un modelo o representación esquemática.

Otro refinamiento deseable es el "suavizar" las curvas. Una línea curva se determina con sólo tres puntos: los dos extremos y el punto más alejado de la línea recta que une ambos. Desde luego, una curva compleja (una S, por ejemplo) necesita ser descompuesta en sus componentes simples para así ser tratada adecuadamente. En este sentido es esencial disponer de algún sistema sencillo que indique al ordenador que la línea en cuestión es una curva que requiere ser suavizada y que no se trata sólo de una línea con un ángulo recto.

Otra característica es la capacidad de introducir luces y sombras en el dibujo. Ante todo es necesario especificar la posición de la fuente luminosa. La parte del objeto dibujado que queda frente al foco luminoso será más brillante y se irá oscureciendo progresivamente para ayudar a definir la forma del objeto. Un software sofisticado permitirá emplear más de una fuente luminosa y tratar la reflexión de la luz de un objeto a otro.

El uso del color tiene directa relación con el som-





Caja de colores mágica
Hasta hace poco tiempo, el método de realización de vistas fijas y secuencias de dibujos animados para el cine y la televisión era muy parecido al que emplean las revistas: el dibujo se hacía sobre papel o una película transparente y luego era fotografiado. El sistema Quantel's Paint Box, sin embargo, elimina por completo el uso del papel componiendo el trabajo artístico digitalmente en el interior del ordenador y grabándolo directamente en la cinta de video

breado. Incluso el más simple de los ordenadores personales dispone en la actualidad de ocho o incluso dieciséis colores, pero los ordenadores gráficos para uso profesional proporcionan, como mínimo, 4 096. Algunos de ellos están limitados simplemente por el número de dígitos binarios del lenguaje del ordenador. Si éste es de 24 bits, por ejemplo, el ordenador tiene unas 16,7 millones de opciones de color. Los mandos de sombreado y coloreado están combinados en uno solo.

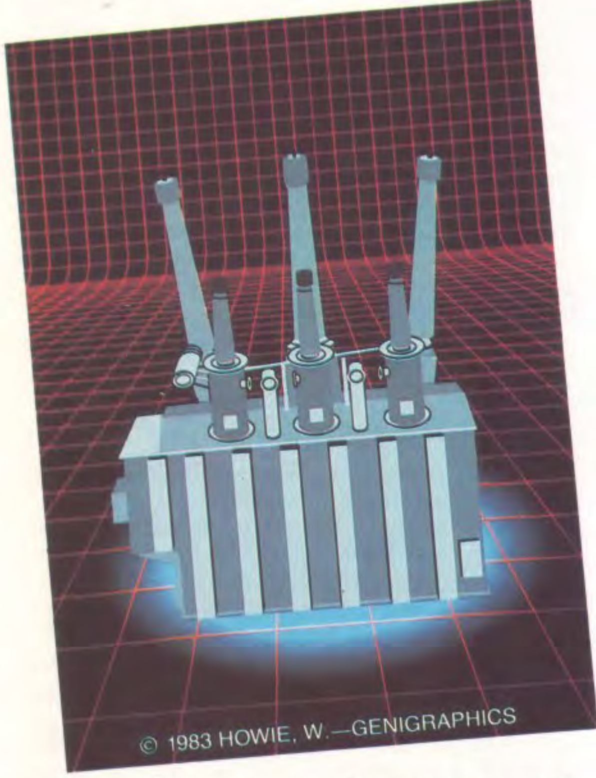
Veamos ahora el problema de simulación de movimiento. Resulta relativamente fácil dividir el movimiento de un objeto en sus componentes, si se le imagina como un problema de geometría de trazos continuos, incluso cuando el objeto representado es tan complejo como una mano humana. El factor determinante es el tamaño y la potencia del ordenador que se está utilizando. Hay que tener en cuenta, sin embargo, que para producir una imagen de alta calidad se necesitará un monitor capaz de dar una resolución de 1 000 × 1 200 pixels. Cada uno de estos pixels requerirá, como mínimo, un byte de ocho bits para tratar la información que define su color y brillo. Esto significa más de un megabyte por pantalla. En consecuencia, la generación de imágenes en movimiento de alta calidad no es posible en los ordenadores personales. La verdad es que los profesionales de este campo utilizan algunos de los ordenadores más grandes y potentes del mundo, y sus honorarios reflejan este hecho, pues un segundo de película acabada puede costar más de 200 000 pesetas.

Si se toma como punto de partida un objeto sencillo, semejante a un cubo, es relativamente fácil entender cómo se puede lograr que se mueva por la pantalla. Un cubo puede definirse sólo por las coordenadas de sus ocho vértices, pero exactamente el mismo principio se aplica a objetos más complejos. La única diferencia está en la cantidad de memoria necesaria para almacenar todas las coordenadas y la potencia de procesamiento que se precisa para poder manejar esa información con la suficiente velocidad para generar movimiento a "tiempo real". En esta aplicación, al igual que en las demás, existe la inevitable interdependencia entre calidad, cantidad de espacio disponible y resolución. Cuanto más pequeña sea la unidad de dibujo que se define, mayor será la necesidad de almacenamiento. Pero, para conseguir una alta calidad de reproducción, es esencial trabajar con el mayor grado de detalle posible.

Un televisor doméstico —con unos 625 × 1 000 pixels— produce imágenes con un "grano" más grueso

que el monitor de alta resolución del que se ha hablado anteriormente. Por tanto se puede estar seguro de que cualquier trabajo con esta o mayor definición será tan realista como una película corriente para televisión. Incluso, con técnicas adecuadas, se puede crear una impresión fiel de realidad mediante dibujos animados.

Para lograr una imagen con este grado de detalle se necesita un software sofisticado así como un hardware adaptado especialmente o fabricado para este fin. El método más popular usa un dispositivo denominado *digitalizador de bits*, el cual es bastante parecido a un gran tablero de dibujo que contiene un cuadriculado alámbrico. Esta malla se emplea para percibir la posición de un cursor que se desplaza a través de ella. Con un digitalizador de bits es posible realizar diversos cometidos: desde reproducir una obra de arte hasta dibujar a mano alzada o utilizar instrumentos de dibujo





convencionales, de la misma forma como si se estuviera dibujando sobre papel. La imagen es digitalizada (son determinadas sus coordenadas X-Y), escrita en la memoria y visualizada por el ordenador. El tipo del trazo que aparece en la pantalla puede definirse tal como se puede elegir entre usar un lápiz, una pluma o un pincel. Asimismo, puede definirse el color haciendo aparecer la paleta, que consiste en una gama de colores localizada en la parte inferior de la pantalla que se asemeja a la paleta de un pintor. Si el color que se desea no es estándar, puede mezclarse exactamente igual que si se usara pintura. El cursor también puede emplearse como goma de borrar, y los "dibujos" pueden superponerse.

Por tanto, habiendo creado una sola imagen, ¿cómo puede lograrse que se mueva? Simplemente mecanizando el proceso convencional, mediante un sistema de ordenador para clasificar el material, colorear siluetas e, incluso, para mostrar el resultado aproximado de una secuencia. Sin duda alguna, este método acelerará el trabajo, pero las técnicas de programación inteligente hacen posible lograr mucho más. Al igual que las curvas pueden ser suavizadas, también es posible realizar los bloques completos de acción, especificando el primer y el último fotograma de una secuencia. Este proceso, en un estudio de animación convencional, es llevado a cabo por un ayudante. Por supuesto, la mayor parte del trabajo de animación es realizado por ayudantes, y es a éstos a quienes sustituye el ordenador.

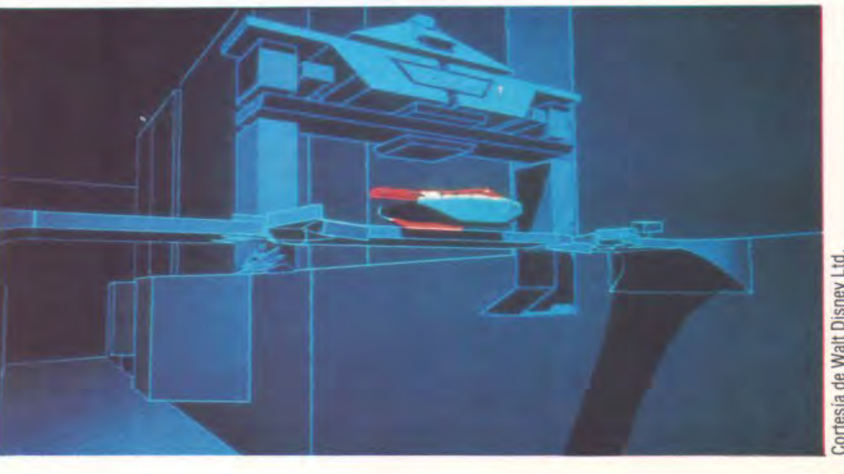
Anteriormente se puso de relieve que la introducción de los ordenadores en el proceso de animación deja libre al artista para que concentre su atención en la calidad de la imagen. La mayor parte del esfuerzo de un realizador de dibujos animados se destina a la labor de crear la ilusión de movimiento, pero esta tarea se puede definir con absoluta precisión en un ordenador.

Una vez establecidas las reglas, simplemente obedeciéndolas se producirá el resultado deseado, que constituirá, una vez más, una segura muestra de que el trabajo que se ha llevado a cabo es adecuado para las técnicas informáticas.

Nuestra pretensión con este artículo sobre la animación ha sido únicamente exponer otra de las innumerables aplicaciones del ordenador. No descartamos, sin embargo, abordar nuevamente el tema con mayor profundidad.

Ver es creer
Gracias al poder de procesamiento ultrarrápido y a la enorme capacidad de almacenamiento de los ordenadores modernos, es posible crear, en el cine o en la televisión, una imagen que no se puede distinguir de una fotografía. Luego, empleando técnicas de programación desarrolladas para la resolución de problemas estadísticos y numéricos, es posible manipular de tal forma estas imágenes creadas, que el espectador cree que son reales

Película pionera
TRON, de la Walt Disney Productions, fue el primer largometraje que empleó técnicas de generación de imágenes con ayuda de ordenador. Combinación de fotogramas rodados convencionalmente y de otros realizados en el interior de un ordenador gigante, esta película mezcló imágenes animadas por ordenador y fotografía de efectos especiales para crear un mundo de fantasía



La traducción alternativa

Los ordenadores "piensan" según el código máquina; los programadores prefieren escribir en un lenguaje de alto nivel como el Basic. Compiladores e intérpretes ofrecen métodos de traducción que difieren de unos a otros

Al principio los ordenadores no tenían teclado. Las instrucciones del programa tenían que introducirse paso a paso situando cada uno de los ocho interruptores en las posiciones *up* o *down* para representar una sola operación. Estas dos combinaciones constituían el código de la máquina.

Era lógico, entonces, reemplazar los interruptores por un teclado tipo máquina de escribir y sustituir el código por palabras de uso corriente. El resultado fue el lenguaje de "alto nivel", el BASIC, por ejemplo, que ha desplazado a los códigos de lenguaje máquina de bajo nivel.

Como procesadores, sin embargo, los ordenadores no cambiaron, sino que continuaron funcionando con los interruptores originales (y aún lo siguen haciendo). Por ello los programadores tuvieron que desarrollar software escritos en la notación original de bajo nivel para traducir esos programas de alto nivel a una forma que los procesadores pudieran manipular. Estos programas de bajo nivel recibieron la denominación de *compiladores* o *intérpretes*, de acuerdo a su método de traducción.

En informática (como en cualquier otra disciplina), todo aumento de potencia o velocidad debe pagarse —en dinero, tiempo o libertad de acción—. Así sucede con los intérpretes y compiladores. Conjuntamente suministran todas las traducciones de programa que se necesitan. Los intérpretes son más adecuados para unas áreas y los compiladores para otras, pero cada uno paga por sus ventajas con desventajas compensadoras.

Los intérpretes, que normalmente se montan en los ordenadores personales, son la forma más barata de traducir programas de alto nivel a un lenguaje que el ordenador pueda entender. No emplean mucha memoria y dejan más espacio para los programas.

Los micros de precio más asequible utilizan casi siempre un intérprete BASIC: se escribe un programa en BASIC, se teclea RUN, y el programa funciona o se detiene e indica un error del sistema, algo parecido a:

```
ERROR DE SINTAXIS EN LINEA 123
```

Entonces de digita LIST, se halla el error, se corrige, se pulsa RUN y el programa funcionará o se volverá a parar, y así sucesivamente. Hay que tener en cuenta que algunos de los intérpretes BASIC más sofisticados verifican los errores de sintaxis al introducir cada línea.

Lo anterior puede haberse realizado cientos de veces sin haber pensado ni un solo momento en el intérprete. Su virtud principal es precisamente el ser un dispositivo invisible que permite trabajar en el programa sin tener que preocuparse de en qué lugar de la memoria se encuentra localizado o cómo ejecutarlo: el programa está en la punta de los dedos, y puede procesarse, listarse o editarse de inmediato.

El intérprete suele ser fácil de usar, siempre que no sea muy sofisticado: cada vez que se digita RUN, el intérprete tiene que encontrar en la memoria el programa en BASIC, traducirlo y realizarlo línea a línea. Si el programa contiene este bucle:

```
400 LET N = 0
500 PRINT N
600 LET N = N + 1
700 IF N < 100 THEN GOTO 500
```

el intérprete debe traducir y realizar las líneas 500 a 700 cien veces, como si nunca las hubiera procesado.

Los compiladores son diferentes. Son caros, difíciles de escribir, y ocupan y usan mucha memoria. Por lo general forman parte de un software soportado en discos; por tanto se necesita un sistema caro.

Lo que ofrecen es flexibilidad, potencia y velocidad; frente a las cuatro líneas de BASIC anteriores, un compilador las traduciría de una vez, luego realizaría ese código cien veces.

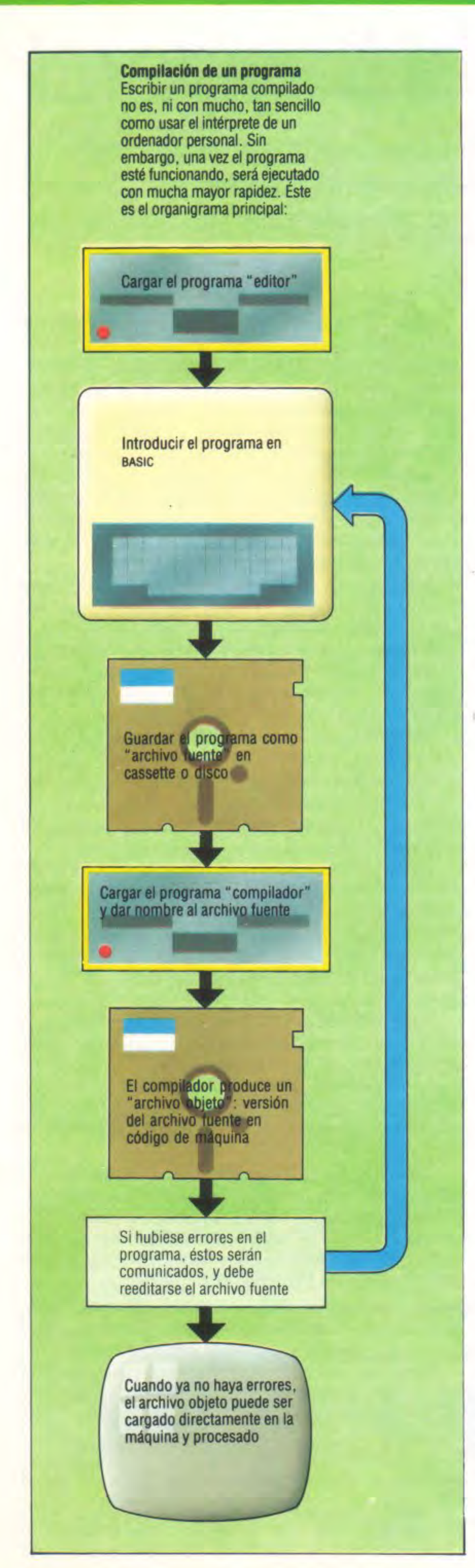
Esto permite ahorrar bastante tiempo, pero tiene un precio. Supóngase que se posee un compilador BASIC y se quiere introducir y procesar un programa en este lenguaje.

Primero se carga y se pasa un programa de creación de archivos (llamado *editor*), que permite introducir el programa mediante el teclado y guardarlo en disco como *archivo fuente*.

A los archivos se les debe dar un nombre para poder encontrarlos una vez se los ha creado (al igual que sucede con un archivo clásico); por lo tanto, el programa editor pide que se les dé una denominación. Estos nombres por lo general constan de dos partes: la primera es una etiqueta, cualquier nombre que se elija (por ejemplo, MYPROG), y la segunda normalmente es un código de tres letras que indica la naturaleza de los datos contenidos en el archivo; este código se denomina *extensión*. En BASIC, podría tener el código BAS. El archivo fuente está ahora en disco bajo el nombre de MYPROG. BAS. Entonces, al escribir el usuario:

```
COMPILE MYPROG. BAS
```

hará que el ordenador cargue y ejecute el compilador BASIC con un archivo denominado MYPROG. BAS.



Compilación de un programa
Escribir un programa compilado no es, ni con mucho, tan sencillo como usar el intérprete de un ordenador personal. Sin embargo, una vez el programa esté funcionando, será ejecutado con mucha mayor rapidez. Éste es el organigrama principal:

El compilador tarda unos segundos, según sea su longitud, en traducir el programa a un *archivo objeto*, que guarda un disco bajo el nombre de MYPROG.OBJ. La extensión OBJ indica que éste es un archivo objeto, es decir, una traducción a código de máquina de un archivo fuente.

Mientras el compilador traduce el archivo, verifica asimismo sus errores de sintaxis. Si encuentra alguno, enviará un mensaje similar a éste:

```
100 REED X:IF X=3(N + 2) LET P=Q
      1          2        3
FATAL ERROR:–
1) //REED// UNRECOGNISED COMMAND
2) //(//ILLEGAL OPERATOR HERE
3)??'THEN' OR 'GOTO' EXPECTED HERE
```

Este mensaje se produce para cada línea que contiene un error. En otras palabras, la información sobre el error es mucho más extensa que la que proporciona un intérprete BASIC.

Llegados a este punto, hay que volver a cargar y pasar el editor, extraer del disco el archivo fuente, realizar los cambios e intentar compilar otra vez. Si no hay más errores, se puede escribir:

```
RUN MYPROG
```

y el programa funcionará, tal como se espera, o no. En esta etapa ya no hay ningún error de sintaxis, porque el programador ya los ha corregido; no obstante, puede darse el caso de que éste quisiera introducir cambios en el programa, para realizar lo cual este último se carga y se pasa el editor, se cambia el archivo fuente, se recopila... y así sucesivamente.

Las virtudes de un compilador no son evidentes en la fase de desarrollo del programa, aunque las informaciones sobre errores son valiosas. Éstos empiezan a adquirir su real importancia cuando ya se tiene un programa de trabajo y se digita la orden RUN.

Los programas compilados son rápidos. Suelen serlo de 5 a 50 veces más que los programas por intérprete, según sea la eficacia del compilador, aunque la velocidad de ejecución del programa compilado tiene su contrapartida en la fase del desarrollo del programa.

Si se comparan compiladores e intérpretes con respecto a las secuencias descritas más arriba, los compiladores quedan en desventaja, puesto que éstos, por lo general, están escritos para máquinas más potentes y menos especializadas, en las cuales se pueden escribir y procesar los programas en otros muchos lenguajes.

El COBOL (lenguaje de alto nivel, para escribir programas de proceso de datos comerciales para control de cuentas, nóminas e inventarios), por ejemplo, fue inventado pensando en utilizar un compilador, mientras que el BASIC requiere realmente un intérprete.

Una vez se ha desarrollado y compilado un programa, el archivo fuente ya no se necesita más que como referencia. En consecuencia, el programa fuente puede ser totalmente comentado y escrito con el propósito de que sea legible, mientras que el archivo objeto puede ser mucho más pequeño y ocupar menos espacio en el disco y la memoria.

El hecho de que los archivos objeto creados por un compilador se compongan de código de máquina ilegible, puede ser, sorprendentemente, una ventaja. Si se comercializa software, no se venderá el archivo fuente, sino sólo el archivo objeto, lo cual hará que sea mucho más difícil hacerlo objeto de piratería, copiarlo o alterarlo.



Lento
En un juego de aventuras, la velocidad no es primordial, y la mayor parte del programa consiste en la manipulación de un texto. Éste está escrito en BASIC, y es interpretado mientras se procesa



Más rápido
Muchos programas de gestión (en particular, hojas electrónicas) son difíciles de escribir en código máquina, debido a que contienen muchas operaciones matemáticas. Sin embargo, un lenguaje de intérprete sería demasiado lento. Por ello, frecuentemente, están escritos en BASIC y luego son compilados



El más rápido
Para los juegos recreativos de acción rápida, que requieren gráficos, incluso un programa compilado sería moroso. Estos paquetes deben escribirse directamente en código de máquina: una tarea lenta y engorrosa

Reproducir el habla

Formaba parte de la ciencia-ficción. Pero, en la actualidad, con un sintetizador de voz, un ordenador puede verdaderamente hablar con usted

Mientras la ciencia de la identificación de voz todavía tiene que desarrollarse totalmente, la generación del habla electrónica ya es una realidad. Hasta hace poco tiempo, sin embargo, los factores esenciales para producir una pronunciación semejante a la humana eran la potencia del ordenador y la capacidad de memoria. Hoy en día, con la ayuda de complementos adecuados, casi todos los ordenadores personales y juguetes electrónicos son capaces de contestar.

Hay dos formas de generar electrónicamente sonidos semejantes a la voz humana. El primero, hasta hace poco tiempo el más común, es el de la síntesis mediante normas. Analizando las frecuencias contenidas en el habla, es posible idear un sistema de reglas que permita crear, a partir de sus componentes, cualquier sonido dado. Por ejemplo, la palabra *mar* podría definirse por el número de milisegundos de la mezcla de frecuencias que crean el sonido m, seguido inmediatamente por las frecuencias de a y r.

Estos bloques de formación individuales se llaman *fonemas* y mediante combinaciones de ellos se puede construir cualquier palabra. Al generar la voz a través de este procedimiento, las características propias de la persona que habla tienden a perderse, pero las palabras pueden reconocerse y entenderse claramente. Debido a que las reglas para generar los fonemas están formadas en el interior del equipo, el usuario puede introducir una lista de los fonemas en el sistema. Éstos son reproducidos por medio de un pequeño altavoz. Con un poco de práctica es posible generar frases completas inmediatamente utilizando secuencias de fonemas, las cuales, por lo general, son susceptibles de ser almacenadas en órdenes de BASIC.

El segundo método de síntesis de voz se basa en la capacidad que poseen el oído y el cerebro para completar los intervalos. Por ejemplo, la gama de frecuencias que puede transmitirse por una línea telefónica sólo proporciona una quinta parte de la calidad que podría esperarse de un equipo corriente de alta fidelidad; no obstante, la voz que se oye a través del auricular es perfectamente inteligible. Ello se debe a que nuestro cerebro completa los intervalos.

El método de síntesis denominado *voz digitalizada* se sirve del mismo fenómeno. Con la reducción de coste de la memoria de ordenador, ahora se puede convertir el habla en información digital mediante un convertidor analógico-digital. Luego los datos resultantes son comprimidos cientos de veces y almacenados en una ROM, creando de ese modo los intervalos que después puede compensar el oído.

Para lograr que cualquiera de las palabras almacenadas sea pronunciada, sólo hay que proporcionar al ordenador la dirección de esa palabra en la ROM, y la información digital es recuperada y convertida otra vez en sonido. Debido a que se han almacenado las palabras originales de la persona que hablaba, se mantienen las características personales de la voz.

Algunos ordenadores poseen hardware y software en discos que permiten al usuario digitalizar su propia voz empleando un micrófono. Los datos resultantes se almacenan en disco (un segundo de conversación ocupa alrededor de 1 Kbyte) y pueden extraerse, cuando sea necesario, de un programa de aplicaciones en forma de mensajes verbales o avisos.

Los usos de los sintetizadores de voz son tantos y tan variados que es prácticamente imposible proporcionar una lista completa de ellos. Para empezar, pueden reemplazar los avisos grabados en cinta magnetofónica de las estaciones de ferrocarril, aeropuertos y otras terminales.

Las unidades de síntesis de voz han sido incorporadas en algunos modelos de automóvil como un componente estándar del tablero de instrumentos. Y esto no es sólo una estratagema de ventas, pues el sintonizador avisa realmente al conductor y éste puede tomar las medidas necesarias sin tener que apartar la vista de la carretera.

En el mercado de los ordenadores personales y de los juegos electrónicos, los sintetizadores de voz se emplean para mejorar sus características: para anunciar la puntuación, para avisar de ataques del enemigo, permitiendo que el jugador se concentre en las tácticas en vez de tener que consultar mensajes impresos en la parte inferior de la pantalla.

Por último, existen dispositivos didácticos tales como *Speak and spell* (Hable y deletree), que recita una palabra que debe ser deletreada correctamente, y diccionarios de idiomas que pronuncian las palabras al aparecer en la pantalla. Más avanzada la obra, trataremos de nuevo el tema con mayor detalle.

El flujo del sonido
La voz puede ser digitalizada y almacenada en memoria, tanto en RAM como en ROM. La señal eléctrica de salida de un micrófono pasa a través de un convertidor analógico-digital. La salida de este chip es en forma digital de unos y ceros. La voz puede recrearse mediante un convertidor digital-analógico, un amplificador y un altavoz

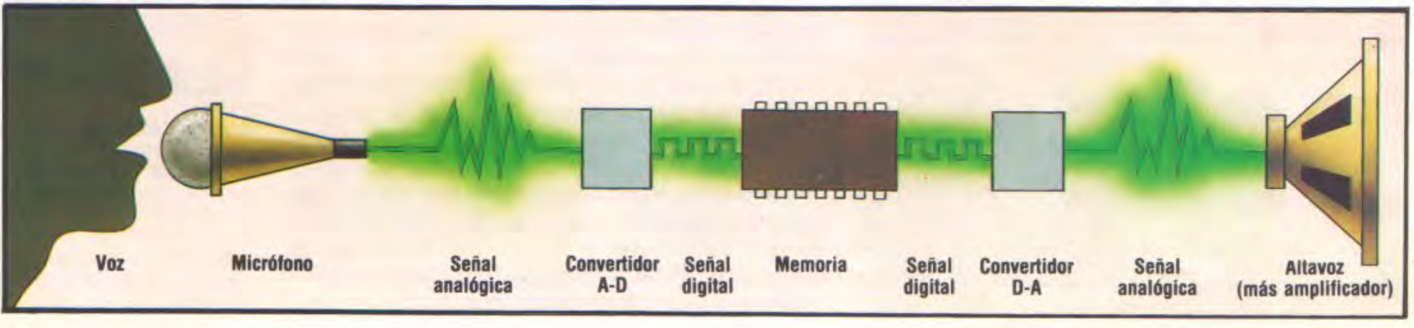


Imagen tridimensional

El cine y la televisión han popularizado en estos últimos tiempos las imágenes en tres dimensiones. Un ordenador puede crear visualizaciones muy eficaces en este sistema

La mayor parte de la gente cree que para que la televisión tridimensional se convierta en realidad antes se habrá de producir algún descubrimiento trascendental. Y esto no es exacto. Con un ordenador es posible producir imágenes tridimensionales de calidad bastante aceptable en un televisor en color corriente.

Eso sí, estas imágenes no son totalmente en color; sólo se visualizan los bordes y las esquinas de un objeto, pero se podrá percibir una verdadera profundidad.

Debe dejarse claro que el término *tridimensional* se aplica con frecuencia para describir programas disponibles a nivel comercial. Pero en la mayoría de los casos este adjetivo no es empleado apropiadamente: el producto que se ofrece es en realidad un juego "plano en pantalla" común y corriente, al que se le ha dado cierta perspectiva o agregado algún sombreado.

Sin embargo, la genuina visualización tridimensional incorpora una tercera dirección de movimiento, hacia afuera y hacia adentro. Esto se consigue utilizando un par de gafas bicolor, una de cuyas lentes es roja y la otra de color azul verdoso. El sistema en tres dimensiones se ha empleado en ciertos libros de ilustraciones y en algunas películas espectaculares de reciente producción como *Jaws-3 (El gran tiburón)*.

La forma de saber si se está mirando una imagen tridimensional falsa o auténtica es a través del color. Las verdaderas visualizaciones tridimensionales no tienen mayor sentido si se las mira sin las gafas, en cuyo caso consistirán mayormente en una serie de líneas trazadas en dos colores diferentes, rojo y azul verdoso. En conjunto, estas líneas constituirán una confusa mezcla de intersecciones. Una imagen que contenga cualquier color menos estos dos, y en particular una imagen que se pueda apreciar con facilidad, probablemente no sea una auténtica imagen tridimensional. Incluso aunque en su etiqueta diga que lo es, el juego carecerá de una verdadera profundidad. El movimiento sólo será posible en dos planos: de arriba abajo y de derecha a izquierda.

Es bastante sencillo producir una imagen que convenza a sus ojos y a su cerebro de que está mirando una escena auténticamente tridimensional, si bien para ello han de entrar en juego las matemáticas.

El primer requisito consiste en proporcionar una imagen ligeramente diferente para cada ojo. Dado que los ojos verán la escena desde posiciones alejadas, las dos imágenes tienen que dibujarse con unos ángulos y una perspectiva ligeramente distintos. Estas representaciones no deben interferirse, aunque ambas estén en la misma pantalla. Esto se logra dibujando una imagen en rojo y la otra en azul verdoso.

Si la imagen se contempla a través de un par de gafas bicolor en las que el filtro de color azul verdoso esté colocado frente al ojo izquierdo, la imagen roja (dirigida al ojo derecho) será casi invisible para el ojo izquierdo. El mismo principio pero a la inversa se aplica para la imagen azul verdosa (dirigida al ojo izquierdo). Mediante este proceso podemos aislar las dos imágenes. El resultado será una imagen tridimensional bastante convincente. Este sistema ofrece el inconveniente de que, aunque hemos utilizado el color para aislar entre sí las dos representaciones, la escena es efectivamente monocromática. Al agregarle rojo al azul verdoso se produce una imagen más o menos blanca, que es lo que su cerebro percibe.



Tony Sleep

Juegos en relieve
El británico Mike Singleton se ha pasado tanto tiempo usando gafas con colores rojo y azul verdoso mientras desarrollaba sus juegos tridimensionales, que quizá ahora para su cerebro sea normal procesar esas imágenes coloreadas. En cualquier caso, Singleton verdaderamente ha dado una nueva "dimensión" a los juegos para ordenadores personales al aplicar a los gráficos una técnica del cómic que ya tiene 30 años de existencia

Otro pequeño problema se hace evidente si consideramos lo que sucedería cuando un objeto se haya de mover "hacia dentro" o "hacia fuera" de la pantalla. Un objeto situado a una distancia de tres metros no aparecerá con un tamaño dos veces más grande que otro que esté a una distancia de seis metros. Los tamaños se deben calcular mediante trigonometría, lo cual se puede realizar empleando la función tangente (TAN en la mayoría de las versiones de BASIC).

El cálculo de los valores y su conversión en coordenadas para el trazado en pantalla supone diversos cálculos con números decimales. Esto es bastante lento y por ello hay un límite respecto al nivel de complejidad que puede alcanzar una visualización o a la rapidez con que ésta se puede mover. La visualización de muchos objetos y la realización de los consiguientes cálculos pueden retardar mucho el proceso, hasta el punto de que el movimiento se perciba espasmódico y lento. Además, para aislar las dos imágenes una debe estar separada de la otra por al menos un punto, de modo que ambas puedan ser de distinto color.

Un factor importante en la producción de imágenes tridimensionales es la calidad de la visualización en pantalla. Se logran mejores resultados con un monitor que con un televisor, pues en el primero los colores son más claros y con márgenes más nítidos.

Peek y Poke

Estas dos órdenes se utilizan cuando se quiere programar algo a lo que no puede hacer frente el lenguaje Basic, pero cada máquina las emplea de modo diferente

POKE
Es necesario utilizar con cuidado la orden POKE, puesto que cambia el contenido de las localizaciones de memoria, y ello podría afectar al funcionamiento del ordenador, y causar la pérdida de un programa. Se puede probar con las siguientes órdenes "seguras" POKE.
En el Atari 400 u 800, si mediante POKE se introduce 1 en la posición 751, el cursor de la pantalla desaparecerá. Pruebe POKE 751,1.
En el Commodore 64, pruebe POKE 1024,1. 1024 es la dirección de la primera localización de pantalla.
En el Spectrum, probar:

```
100 FOR N = 0 TO 6
    STEP 2
110 POKE USR "A" + N,
    BIN01010101
120 POKE USR
    "A" + N + 1,
    BIN10101010
130 NEXT N
140 PRINT "AAAAAAAA"
```

Las A de la línea 140 deben introducirse a través del módulo de gráficos. Al procesar el programa se producirá una línea de símbolos de tableros de ajedrez en miniatura. Sin embargo, también puede producir formas de interferencia en el televisor

PEEK y POKE son dos órdenes del lenguaje BASIC que se emplean en programas avanzados, que necesitan manipular bits y bytes individuales en la memoria. La orden PEEK se usa para examinar el contenido de una dirección específica (localización) de la memoria, y la orden POKE se utiliza para almacenar un número (del 0 al 255) en un punto específico de la memoria.

Mediante ellas el programador puede acceder al funcionamiento interno del ordenador de un modo que no sería posible con otro sistema. Normalmente el BASIC propio del ordenador es el que se ocupa de saber los puntos donde se encuentran almacenados los datos y las variables que definen los caracteres a visualizar en la pantalla. Aunque, por regla general, el operador no debe preocuparse en saber en qué parte de la memoria se hallan localizados estos datos, a veces sí puede ser necesario. La orden PEEK permite hacerlo.

Fácilmente puede escribirse un programa corto para examinar cualquier punto de la memoria:

```
10 REM BUSCANDO LOCALIZACIONES DE MEMORIA
20 PRINT "INTRODUCIR LOCALIZACION DE MEMORIA
   EN DECIMAL"
30 INPUT M
40 P = PEEK(M)
50 PRINT "CONTENIDO DE LOCALIZACION";M;
   "ESTA";P
60 GOTO 20
70 END
```

Esto imprimirá el contenido de la dirección específica, expresada en forma de número decimal. (En realidad, por supuesto, el ordenador lo almacena en binario.) Si se quiere ver a qué es equivalente el contenido en términos de caracteres "imprimibles", el lenguaje BASIC incluye una función que convierte los números decimales en sus caracteres equivalentes. Ésta es la función CHR$, y cambiando la línea 50 se imprimirán los equivalentes de los caracteres:

```
50 PRINT "CONTENIDO DE LOCALIZACION";M;
   "ESTA";CHR$(P)
```

Para examinar la totalidad de la memoria, puede añadirse un bucle FOR...NEXT, suprimiendo la línea 30, cambiando la línea 20 por FOR X = 0 TO 65535 y sustituyendo la línea 60 por NEXT X.

Si se quiere tener tiempo suficiente para ver cuándo se imprime cada carácter, quizá sea necesario añadir un bucle de retraso tras la orden PRINT y antes de NEXT X. Téngase en cuenta también que el límite superior del bucle FOR...NEXT presupone que se tiene una memoria de 64 Kbytes. Este número puede cambiarse para memorias más pequeñas: 16 Kbytes requiere, en notación decimal, 16383; 32 Kbytes necesita 32767, y 48 Kbytes precisa 49151. El desarrollo completo de este programa es:

```
10 REM PEEKING AND PRINTING TODAS LAS
   LOCALIZACIONES
20 FOR X = 0 TO 65535
30 LET Y = PEEK(X)
40 PRINT "LOCALIZACION";X;"=";Y;"=";
50 PRINT CHR$(Y)
60 FOR D = 1 TO 200
70 NEXT D
80 NEXT X
90 END
```

Aunque la función CHR$ convierte los números decimales en sus equivalentes, los caracteres imprimibles se representan con los números 32 a 127. La mayor parte de los ordenadores utilizan los números entre 128 y 255 (el número mayor representable en un solo byte) para caracteres gráficos especiales.

Muchos de los números entre el 0 y 31 desempeñan funciones especiales de control de pantalla. Al pasar el programa, cuando éstos son encontrados en la memoria, serán convertidos, por la CHR$, en efectos de pantalla especiales, por ejemplo, harán que la pantalla se vuelva negra o que el cursor se desplace a la esquina superior izquierda.

La orden POKE es, en esencia, opuesta a PEEK. Permite "escribir" un byte o dato (cualquier número del 0 al 255) en un punto de la memoria. Debe tenerse cuidado al emplear POKE; si se utiliza para introducir un número y se hace en una sección errónea de la memoria, podría destruir o alterar parte del programa esencial. La única forma de remediar este hecho sería arrancar de nuevo el ordenador (desconectándolo y volviendo a conectarlo otra vez, a menos que tuviera un mando de borrado), y esto comporta el riesgo de destruir alguno de los programas. Por consiguiente, antes de utilizar la orden POKE, se debe consultar el manual para hallar en el esquema de la memoria un área designada como *área para el usuario*.

En muchos ordenadores personales, la memoria video (la empleada para almacenar los caracteres que se visualizarán en la pantalla) es accesible al usuario. Normalmente, el ordenador recibe la información sobre la forma de los caracteres a visualizar desde un circuito ROM especial, llamado generador de caracteres, el cual almacena las estructuras de puntos de cada carácter. Pero, por lo general, también es posible emplear una RAM. Si los códigos de las estructuras se almacenan en RAM, se podrán introducir, mediante la orden POKE, nuevas estructuras, expresadas como números decimales, en las direcciones apropiadas de la RAM, y usarlas para definir caracteres visualizables totalmente nuevos.



TI99/4A

El ordenador personal de la Texas Instruments es como un Mercedes entre Volkswagen. La calidad de construcción es elevada, pero los complementos son caros

Desde el punto de vista del diseño y construcción, el TI99/4A de la Texas Instruments es uno de los ordenadores personales más profesionales. Es una máquina que tiene tras de sí la experiencia de una firma con un prestigio reconocido hace mucho tiempo en el mundo de la informática.

Utiliza un microprocesador de 16 bits, el TMS9900, diseñado y fabricado por la misma Texas Instruments, compañía que también fabrica semiconductores, calculadoras, microprocesadores y miniordenadores. El microprocesador TMS9900 fue uno de los primeros chips de 16 bits, pero no logró obtener una amplia popularidad.

El TI99/4A tiene un teclado con 48 piezas, que, comparado con el nivel general de los ordenadores personales, resulta muy cómodo. A su derecha existe un espacio destinado a los cartuchos de software, denominado por la firma fabricante como software de "estado sólido". Una conexión similar, situada en el borde derecho del aparato, permite adaptar una extensión de hardware. Los módulos de ampliación, que son grandes cajas de plástico, contienen controladores de unidades de disco, ampliación de memoria y una interface serial (RS232). Se conecta por medio de una caja de expansión, unidad considerada esencial si se desea ampliar la máquina.

La pantalla posee 16 colores con gráficos de alta resolución; el generador de sonido es capaz de producir tres notas o "voces" independientes al mismo tiempo. Sin embargo, la falta de buena documentación hace que sea difícil aprender a escribir los programas en código de máquina, el uso de gráficos y la utilización de las características de sonido.

Texas Instruments vende directamente sus equipos, así como a través de distribuidores, y prácticamente sólo existen los periféricos que suministra la misma firma.

El ordenador TI99/4A está diseñado para poder ser utilizado por principiantes en informática. Su lenguaje es el BASIC, y el más popular como complemento es el LOGO. En Estados Unidos es muy empleado en los centros escolares, compitiendo con el Apple II para el primer puesto en el concepto de ventas entre los ordenadores didácticos.

Cuando se conecta, en la pantalla aparece un menú que ofrece al usuario un número variado de eleccio-



Chris Stevens

Teclado del TI99/4A
Es de un nivel más elevado que el de la mayoría de los ordenadores personales, aunque varios usuarios han comentado que el "bote" de cada tecla es demasiado duro. El número de teclas es también bastante limitado, presumiblemente para dejar sitio a la carga de cartuchos, en el lado derecho. La mayoría de las teclas, por tanto, tiene dos funciones: pulsando "CTRL" y "E" se logra que el cursor se desplace hacia arriba. La tecla "FCTN" convierte la hilera superior de teclas en definibles por el usuario, y es posible insertar una cinta plástica sobre esta hilera, en la cual puede escribirse la designación para cada pieza

nes. Si se introduce un cartucho de software en el ordenador, el operador tiene la posibilidad de pasar el nuevo software o el BASIC. Las cualidades del BASIC incorporado en el ordenador son limitadas, pero cabe la posibilidad de adaptar un cartucho BASIC "ampliado", con el cual se enriquecen sus potencialidades (mejorando incluso las que ofrece un Microsoft estándar) y se proporcionan órdenes de impresión normalizadas (véase p. 53), gráficos "sprite" y la posibilidad de manejar un sintetizador de voz. Para el funcionamiento de este último es necesario el BASIC ampliado o un cartucho editor de voz.

El TI99/4A posee numerosas ampliaciones de software y hardware. Existe todo tipo de periféricos y puede adquirirse una variada gama de lenguajes de programación.

El sistema de este ordenador, en su conjunto, es de fácil manejo para los principiantes, y su sólida construcción lo hace adecuado para ser utilizado incluso por niños de corta edad.



La palanca de mando
Las palancas de mando de la Texas Instruments (designadas *controladores por cable*) están formadas por un par de unidades, conectadas a un enchufe para su adaptación al ordenador. En cada dispositivo hay cuatro interruptores, que se asemejan a las conexiones de la parte inferior de algunos teclados



Caja de ampliación periférica
Esta caja dispone de una fuente de energía, conexiones y espacio para los módulos para ampliación de memoria, interfaces de discos e impresora. Estos módulos están formados por cajas de plástico que contienen circuitos con un conector terminal en la base, un testigo óptico de conexión en la parte frontal y diversos cables procedentes de la cara posterior. En la caja hay capacidad para ocho cargas. La primera de la izquierda contiene el módulo que conecta la caja de ampliación al ordenador, y la de la derecha está destinada al módulo electrónico de la unidad de disco. Quedan seis cargas para la memoria y la ampliación de la conexión serial. Sólo se puede añadir un módulo extra de memoria de 32 Kbytes, con una RAM máxima de 52 Kbytes.
El módulo interface serial permite conectar dispositivos en serie como impresoras y modems de formato RS232

Chris Stevens

Conexión video
Esta conexión suministra las señales básicas para la generación de los sistemas de televisión PAL (España y casi toda Europa occidental) y NTSC (EE. UU.)

RAM
La máquina sale de fábrica con una RAM de 16 Kbytes, que puede ampliarse externamente

Conexiones palanca de mando
Esta conexión única de patillas múltiples permite empalmar las dos palancas de mando, fabricadas también por Texas

Componentes discretos
Otra característica de ordenadores semejantes a éste, que fueron diseñados hace varios años, es su gran número de componentes discretos: transistores y resistencias. Actualmente, un chip puede sustituir a docenas de ellos

ROM
La ROM de la máquina puede ser complementada con la conexión de cartuchos. Por ejemplo, al ampliar el BASIC se incrementará el número de mandos disponible

Interruptor
Incorpora un indicador LED de conexión



Conexión cassette
El TI99/4A puede funcionar con dos grabadoras de cassette corrientes, y controlar el motor de una de ellas. Esto significa que puede ejecutar programas comerciales sin acabar, que requieren la copia de datos de una consola a otra

CPU
El TMS9900 es uno de los primeros procesadores, lo cual explica su gran tamaño. Todas las líneas de dirección y datos, así como las líneas de control, tienen conexiones independientes. Los procesadores más modernos poseen patillas para varias funciones y por ello un chip puede tener menor número de ellas. Es un microprocesador de 16 bits

Conexión periféricos
Es sólo un conector terminal PCB, al que se acoplan otras unidades. Texas lo denomina como interface CRU (*Communications Register Unit*: unidad de registro de comunicaciones). Antes de que se introdujera una unidad ampliadora de uso general, cada periférico se conectaba en serie al siguiente en una larga línea

Conexión paquete ROM
Los paquetes ROM, que Texas llama *módulos de mando de estado sólido*, se conectan en este punto. El mecanismo es mucho más sólido que en la mayor parte de las máquinas

Memoria "scratchpad"
Los chips marcados con 6810 constituyen una memoria *scratchpad* ("bloc de apuntes"), esencial para el funcionamiento del 9900. Este microprocesador es diferente de todos los demás al no tener localizaciones de memoria interna (registros) y por ello necesita memoria externa. Ésta no es accesible a los programas normales

TI99/4A

DIMENSIONES
380 × 260 × 70 mm

PESO
1,8 kg

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
ROM: 28 Kbytes; RAM usuario: 16 K; RAM gráficos: 8 K. Existen 256 bytes extra de RAM *scratchpad*, normalmente no accesibles al usuario. Se emplean para registros internos del 9900; la mayoría de CPUs lo llevan incorporado

VISUALIZACION EN VIDEO
Visualiza caracteres en 24 líneas de 32 columnas. Existen 16 colores, que pueden utilizarse como primeros planos o fondos. En la máquina básica no se dispone de gráficos de usuario, pero, con una secuencia de 16 caracteres, pueden definirse celdas individuales de 8 × 8 caracteres

INTERFACES
Cassette, palanca de mando, video (no TV), carga de cartuchos y conexión para el bus de ampliación

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
BASIC ampliado, TI LOGO, UCSD (University of California at San Diego), PASCAL, TI FORTH y Assembler

VIENE CON
Adaptador toma de potencia, adaptador TV, conexión cassette y manuales

TECLADO
Tipo máquina de escribir con 48 teclas móviles, incluyendo teclas de control y función. Según los cartuchos de software que se introducen, las teclas numéricas lo son también de función

DOCUMENTACION
Viene acompañado de un manual principal que describe cómo conectar el ordenador y usar los "módulos de mando sólido". Esta introducción es muy corta y tiene diagramas, pero no fotografías. Hay una lista detallada de las funciones disponibles en BASIC, una sección con ejemplos de algunos programas y un pequeño glosario al final

Piratas en acción

La piratería es una espina clavada en la industria del software, y los distribuidores adoptan severas medidas para combatirla



Copiador esclavo
Las cassettes de software, al igual que las de música, se duplican mediante un copiador de alta velocidad. Consiste en una consola en la cual se coloca la cassette original y en un número de unidades esclavas que efectúan las grabaciones simultáneamente. Copiar ambas caras de una cassette de programa es sólo cuestión de segundos. Los discos tienen que copiarse uno a uno utilizando unidades de disco corrientes

Ciento a uno
Del mismo modo que es ilegal realizar copias de cassettes de música propiedad de otras personas, efectuar copias de programas representa incurrir en piratería de software. Desgraciadamente para los distribuidores, ésta no sólo es difícil de evitar, sino también de detectar y procesar. Algunos fabricantes afirman que por cada ejemplar vendido legalmente se realizan cien copias ilegales

Tony Sleep

La piratería de software puede definirse, simplemente, como la copia de programas no autorizada. Al igual que en la industria del disco, con la cual tiene muchos problemas en común, la piratería se produce de diferentes formas y a distintos niveles. Al nivel más bajo, se comete piratería cada vez que el propietario de un ordenador personal copia un programa que le ha prestado un amigo. Incluso el hecho de que algunos programas (especialmente los escritos en código de máquina) no puedan conservarse con las órdenes normales de BASIC, no constituye un impedimento importante, puesto que siempre será posible conectar entre sí dos cassettes y copiar el programa, sin necesidad de emplear un ordenador.

Algunos distribuidores de juegos afirman que por cada ejemplar vendido, se realizan hasta cien copias ilegales. Aunque se pueda argüir que algunos fabricantes están en condiciones de afrontar estas pérdidas, debe recordarse que muchas personas obtienen su medio de vida gracias a los royalties de los programas, ¡y no viajan en Rolls Royce!

Se ha producido una amplia controversia sobre el hecho de que los distribuidores ofrezcan software con los sistemas de préstamo, alquiler o "pruebe antes de comprar", puesto que éstos facilitan la copia de programas. Algunos vendedores con menos escrúpulos van más lejos y regalan copias piratas de títulos populares a todo comprador de un ordenador, con el fin de incrementar su valor efectivo.

Existen casos incluso de distribuidores que reproducen programas a gran escala y los venden a otros —con menos riesgo del que parece si éstos son de otro

país—. Estos productos son equivalentes, por tanto, a las copias de "contrabando" de álbumes discográficos de gran popularidad.

A partir de este punto, la piratería se vuelve más sofisticada y más difícil de precisar. Por ejemplo, alguien toma un programa que ya existe, efectúa algunas modificaciones en él y lo pone a la venta como propio. La nueva versión puede ofrecer una mejora sustancial en rendimiento o en sus características, o simplemente puede presentar un cambio en los títulos de crédito que aparecen al pasar el programa por primera vez, o bien alterar la disposición de la información en la pantalla para que el paquete no sea reconocible inmediatamente. Este método se utiliza con mayor frecuencia en programas de gestión que en software de juegos.

Se puede discutir si el proceso de modificación de un programa puede considerarse como piratería de software, es decir, como si fuera una simple copia. Ésta es la causa de que mucha gente lleve a cabo esta labor impunemente. Los editores de programas están muy poco protegidos por la justicia y las leyes existentes sobre derechos de copyright no protegen a los programas de la piratería y de las modificaciones. Parece ser que los derechos de copyright sólo son aplicables a material impreso (con excepciones especiales para la música), y, en consecuencia, los programas de ordenador, que únicamente están almacenados en RAM o en cassette, no están amparados por estos derechos. Como en otros muchos temas legales, debe establecerse una jurisprudencia, y esto requiere, lógicamente, tiempo y dinero.

El punto más controvertido radica en el hecho, por ejemplo, de que una compañía tome una idea de un programa popular y reproduzca su propia versión. Nótese que no se está copiando ninguno de los códigos de programa, lo único que se hace es tomar notas precisas de cómo aparece el programa en la pantalla y de cómo reacciona a las órdenes que se le suministran, y luego empezar desde cero para conseguir el mismo efecto. El ejemplo más sobresaliente sobre esto ha sido el caso de PacMan, que empezó por ser un juego recreativo de máquinas tragaperras, fue puesto a la venta por la casa Atari para sus ordenadores personales y en cartuchos de video y posteriormente realizado en diferentes versiones por varios editores de software. Cada una de las versiones era ligeramente diferente, pero todas presentaban como atracción principal a la misma pequeña y familiar criatura que engulle ávidamente su camino alrededor del laberinto. Después de varios meses, Atari consiguió eliminar a la mayoría de estos competidores, ya fuera mediante la acción de la justicia o bien, en casos menos importantes, simplemente con la amenaza de llevar el caso a los tribunales.

Por lo general, los autores y distribuidores de software poseen medios, además de los legalmente establecidos, de proteger sus códigos de programa y royalties. Algunos distribuidores son partidarios del plausible punto de vista de que si venden sus productos a un precio suficientemente bajo, la gente tendrá menos interés en copiarlos. En programas más sofisticados, un manual bien editado o una presentación atractiva aportan un cierto grado de protección.

El "registro del usuario" es un método con el que se protegen softwares de gestión más caros: hasta que no se envía la tarjeta incluida en el manual del usuario, no se ofrece ninguna ayuda o asistencia técnica por vía telefónica.

Los llamados métodos "duros" de protección, normalmente, consisten en un dispositivo del tamaño de una caja de cerillas, denominado *dongle*, que debe ser adaptado a una de las conexiones de las interfaces del ordenador para que pueda procesarse el programa. Los circuitos del dongle incorporan un corto programa electrónico, normalmente un esquema de unos y ceros grabado en una de las ROM. Con cierta frecuencia el programa de aplicaciones se dirige al dongle; si no recibe la contestación del código correcto, rehúsa continuar. El código puede muy bien ser específico para cada dongle, lo cual significa que cada copia del programa sólo funciona con el dongle con el que se vende. La única forma de hacer copias ilegales es fabricar este dispositivo, o volver a escribir el código de programa para suprimir las secciones que hacen referencia a aquél. Esto no es imposible de realizar, pero está más allá de las posibilidades de la mayoría de los programadores personales.

Se han realizado muchas investigaciones con el fin de obtener una protección similar, sin tener que utilizar hardware adicional. La idea, acertadamente conocida como *filigrana*, consiste en tener un código magnético superpuesto en la cassette o el disco, "detrás" de la grabación del propio programa, y que no se transfiere al efectuar una copia. Por tanto el programa sólo funcionará con el disco o cassette originales.

La única protección "dura" económicamente viable que pueden utilizar los distribuidores de juegos por ordenador es el cartucho de ROM, el cual por lo general se vende a precios altos porque evita los largos periodos de tiempo de carga de cassettes. Sin embargo, ni los cartuchos son inexpugnables. En la actualidad existen dispositivos que pueden copiar un cartucho en una cassette, o en una especie de cartucho que el usuario puede programar o reprogramar.

La piratería del software es una batalla del tipo "policías y ladrones" en la cual los protagonistas tratan constantemente de superar en ingenio e inventiva a sus rivales en el mercado. No obstante, parece prácticamente imposible que se pueda eliminar; en el mejor de los casos puede conseguirse que sea lo suficientemente costosa como para convertirla en una actividad marginal.

Dongles
Existen unos pequeños dispositivos de hardware que se utilizan para proteger de la copia ilegal ciertos programas. Éstos no funcionarán a menos que se conecte el dongle adecuado en las interfaces del ordenador. Generalmente, sus circuitos electrónicos están recubiertos con resina sólida, y por tanto son muy difíciles de manipular



Ian McKinnell

Una nueva dimensión

Las matrices unidimensionales, como se ha visto, almacenan cierta cantidad de datos que tienen una característica común. Las matrices bidimensionales, por su parte, se utilizan para confeccionar tablas y cuadros

Hasta ahora se han considerado dos tipos de variables: simples y subíndice. Las primeras son semejantes a posiciones de memoria, en las cuales pueden almacenarse y manipularse números (o series de caracteres) refiriéndose a la variable "etiqueta". Las variables simples únicamente pueden almacenar un valor o serie y tienen nombres de variable "simple" —por ejemplo, N, B2, X, Y3—. Las variables subíndice, a veces llamadas matrices unidimensionales, pueden almacenar una lista entera de valores o series. El número de valores o series que tienen cabida se especifica al principio del programa mediante la orden DIM. Por ejemplo, DIM A(16) significa que la matriz denominada A puede contener 16 valores independientes. Debe tenerse en cuenta, sin embargo, que muchos lenguajes BASIC aceptan A(0) como primer elemento, y por consiguiente DIM A(16) define en realidad a 17 elementos. Para hacer referencia a estas "posiciones" se debe emplear el subíndice apropiado. PRINT A(1) imprimirá el primer elemento de la matriz: LET B = A(12) asigna el valor del decimosegundo elemento de la matriz a la variable B; LET A(3) = A(5) traspasa el valor del quinto elemento al tercero.

Algunas veces, sin embargo, es necesario poder manejar datos cuya mejor forma de representación es mediante tablas. Adviértase que tienen una gran semejanza a una hoja electrónica (véase p. 158). Los datos a representar pueden ser muy diversos, abarcando desde los resultados de los partidos de fútbol hasta la clasificación de ventas de unos grandes almacenes por artículos y departamentos. Como ejemplo de una tabla de datos típica, véase el siguiente desglose de los gastos de una familia durante el período de un año:

	Alquiler	Teléfono	Electricidad	Alimentación	Coche
ENE	20 000	3 800	3 500	52 700	8 300
FEB	20 000	3 500	2 900	51 000	6 800
MAR	20 000	3 400	3 100	49 900	7 000
ABR	20 000	3 300	3 000	50 000	9 100
MAYO	20 000	3 400	4 300	47 000	7 900
JUN	20 000	3 600	3 000	48 600	6 100
JUL	20 000	3 500	3 200	47 000	8 500
AGO	20 000	3 400	3 100	49 500	7 400
SEPT	20 000	3 300	3 000	46 300	6 900
OCT	20 000	3 700	3 800	52 800	7 000
NOV	20 000	3 900	3 800	53 000	7 500
DIC	20 000	4 100	4 700	55 900	8 600

El disponer la información de esta manera permite que sea tratada de forma relativamente simple. Es fácil, por ejemplo, hallar el total de los gastos del mes de marzo con sólo sumar todas las cantidades que figuran en la línea correspondiente a este mes. Igual de sencillo resulta hallar los costes anuales de teléfono y del automóvil sumando las columnas verticales. De forma semejante, no resulta difícil conocer los promedios mensuales o anuales de gastos por dichos conceptos. Esta tabla recibe el nombre de matriz bidimensional. Consta de 12 filas y 5 columnas.

En BASIC, las matrices bidimensionales semejantes a la anterior pueden ser representadas casi de la misma forma que las unidimensionales. La diferencia radica en que, para hacer referencia a su situación, ahora la variable requiere dos subíndices.

Si se estuviera escribiendo un programa en BASIC utilizando esta tabla de información, lo más sencillo hubiera sido tratar toda la tabla como una matriz bidimensional. Al igual que a una matriz ordinaria de una dimensión, se le atribuye un nombre, por ejemplo A. Y al igual que en aquéllas, será necesario DIMensionarlas. Como existen 12 filas y 5 columnas, se dimensionará así: DIM A(12,5). Es importante el orden en que se escriben los dos subíndices; la norma es que se coloquen las filas en primer lugar y las columnas en segundo. La tabla representada más arriba tiene 12 filas (una para cada mes) y cinco columnas (una para cada tipo de gasto), siendo por tanto una matriz de 12 por 5.

La orden DIM desempeña dos funciones esenciales. Deja aparte, en la memoria del ordenador, las posiciones necesarias para la matriz, y permite que cada una de estas posiciones sea determinada por el nombre de la variable seguido, entre paréntesis, por las posiciones de las filas y columnas. La orden DIM X(3,5), por ejemplo, crearía una variable X capaz de representar una matriz con tres filas y cinco columnas.

Véase la tabla y supóngase que la información ha sido introducida como si fueran los elementos de una matriz bidimensional denominada A. Hállense los valores de A(1,1), A(1,5), A(2,1), A(3,3) y A(12,3).

Es posible introducir una tabla de información como si se tratara de una matriz en parte de un programa, empleando la orden LET; a continuación damos un ejemplo:

```
30 LET A(1,2) = 3 800
40 LET A(1,3) = 3 500
50 LET A(1,4) = 52 700
 :
 :
610 LET A(12,5) = 8 600
```

Pero evidentemente ésta es una forma muy laboriosa de realizar el programa. Un procedimiento mucho más simple es emplear las órdenes READ y DATA, o bien INPUT junto con bucles FOR...NEXT intercalados.

Así, pues, veamos cómo podría hacerse utilizando la orden READ:

```
10 DIM A(12,5)
20 FOR F = 1 TO 12
30 FOR C = 1 TO 5
40 READ A(F,C)
50 NEXT C
60 NEXT F
70 DATA 20 000, 3 800, 3 500, 52 700, 8 300,
   20 000, 3 500, 2 900
80 DATA 51 000, 6 800, 20 000, 3 400, 3 100,
   49 900, 7 000, 20 000
90 DATA 3 300, 3 000, 50 000, 9 100, 20 000, 3 400,
   4 300, 47 000
100 DATA 7 900, 20 000, 3 600, 3 000, 48 600,
    6 100, 20 000, 3 500
110 DATA 3 200, 47 000, 8 500, 20 000, 3 400,
    3 100, 49 500, 7 400, 20 000
120 DATA 3 300, 3 000, 46 300, 6 900, 20 000,
    3 700, 3 800
130 DATA 52 800, 7 000, 20 000, 3 900, 3 800,
    53 000, 7 500
140 DATA 20 000, 4 100, 4 700, 55 900, 8 600
150 END
```

En este programa existen varios puntos importantes a tener en cuenta. El primero de ellos es que DIM está justo al principio del programa. Una orden DIM debe efectuarse una sola vez en un programa y, por lo tanto, lo normal es colocarla cerca del principio o antes de realizar ningún bucle. El segundo punto a considerar es que hay dos bucles FOR...NEXT, uno para situar la "fila" y otro para hacer lo mismo con la "columna". Estos dos bucles no están uno a continuación del otro, sino que uno está alojado en el interior del otro. Ténganse en cuenta los límites elegidos. FOR F = 1 TO 12 incrementará el valor de la fila de 1 a 12; FOR C = 1 TO 5 aumentará el valor de la columna de 1 a 5.

Exactamente en el centro del segundo bucle —que está alojado en el interior del primero— está situada la orden READ. La parte decisiva del programa es la siguiente:

```
20 FOR F = 1 TO 12
30 FOR C = 1 TO 5
40 READ A(F,C)
50 NEXT C
60 NEXT F
```

La primera vez que se procese, después de haber realizado las líneas 20 y 30, los valores asignados a F y C serán ambos 1, y, por lo tanto, la línea 40 será equivalente a READ A(1,1). La primera cantidad de la orden DATA es 20 000, y, en consecuencia, este valor será asignado a la primera fila y a la primera columna de la matriz.

Después que haya sucedido esto, NEXT C hace retroceder el programa a la línea 30 y el valor asignado a C pasa a ser 2. Ahora la línea 40 es equivalente a READ A(1,2) y la siguiente cantidad, 3 800, será asignada a la primera fila y segunda columna de la matriz. Este proceso se repite hasta que C haya alcanzado el valor 5. Después de ello, la orden NEXT F de la línea 60 hace retroceder el programa hasta la línea 20 y F toma el valor 2. La línea 30 ajusta otra vez el valor de C a 1 y, por consiguiente, ahora la línea 40 será equivalente a READ A(2,1).

De esta forma, los bucles anidados en esta manera son muy útiles, pero es necesario poner mucha atención. Cada bucle debe estar alojado completamente en el interior del otro y debe observarse cuidadosamente la correlación de las órdenes NEXT. Obsérvese cómo el primer bucle, FOR F, incluye la segunda orden NEXT. Cuando existen dos bucles, uno de ellos alojado en el interior del otro, el primero recibe el nombre de bucle exterior, y el segundo, el de bucle interior. Todo el bucle interior será siempre completado antes de que sea incrementado el índice del bucle exterior. Es posible introducir bucles en el interior de otros hasta la "profundidad" requerida por el programa, pero éste puede resultar complejo y difícil de seguir. Es una mala práctica colocar en el interior de los bucles instrucciones que hagan desviar el programa principal; asimismo debe evitarse emplear GOTO.

Veamos la orden DATA. Obsérvese que, para separar los datos de las cantidades, se emplean comas, pero que no tiene que haber coma antes de la primera cantidad o después de la última. Entre cada cantidad se han insertado espacios, pero esto no es lo normal. Al introducir los datos, es fácil cometer errores, que luego son difíciles de descubrir.

Se pueden utilizar tantas órdenes DATA como sean necesarias. Cada línea nueva debe empezar con una DATA. Cada vez se lee una cantidad, empezando por el principio de la primera DATA y continuando hasta que se hayan leído todas. Compruébese que el número de datos sea correcto, pues, en caso contrario, al pasar el programa se obtendrá una información errónea.

El programa presentado hasta ahora, realmente, no hace más que transformar los datos y disponerlos en forma de una matriz bidimensional. Después de introducir el programa y pulsar RUN, aparentemente no sucederá nada, y lo único que se verá en la pantalla será el esquema BASIC. Para comprobar que los datos están bien situados, deben probarse algunas órdenes PRINT. (En BASIC, una orden es una palabra clave que puede ejecutarse inmediatamente, sin tener que utilizar el programa, y, por tanto, no necesita un número de línea. Por ejemplo, LIST, RUN, SAVE, AUTO, EDIT y PRINT.) PRINT A(1,1) <CR> hará que el número 20 000 aparezca en la pantalla. ¿Qué se imprimirá con las siguientes órdenes?

```
PRINT A(12,1)
PRINT A(1,5)
PRINT A(5,1)
PRINT A(5,5)
```

Para lograr que el programa sea de utilidad, es necesario ampliarlo. En su forma actual constituye una base adecuada para un "programa principal". Para usarlo como parte de un programa mayor y más útil, pueden escribirse módulos como si constituyeran subrutinas, que serán empleadas por diversos GOSUB, insertados en puntos adecuados antes de la orden END.

En los primeros pasos del diseño de un programa de gastos domésticos, es mejor empezar con una simple descripción escrita de los requisitos generales. Puede decidirse que se quiere obtener el total de gastos y los promedios por meses, o bien para cada categoría (electricidad, por ejemplo). Se pueden desarrollar los detalles de cómo derivar estos resultados a una etapa posterior. Si se tiene que efectuar una elección en el programa sobre qué subrutinas se desea que sean realizadas, probablemente se requerirá la guía de un "menú", que controle directamente las subrutinas apropiadas. Un primer bosquejo del programa, en

esta fase de desarrollo, puede ser semejante al siguiente:

```
PROGRAMA PRINCIPAL
(ENTRADA DE DATOS)
MENU
(SELECCION DE SUBRUTINAS)
END
```

Al desarrollar el programa con algo más de precisión, se verá la necesidad de emplear subrutinas para calcular totales por meses (TOTALMES) o por categorías (TOTALCAT), los gastos promedios mensuales (PROMMES) y los promedios anuales por categorías (PROMCAT). La razón de utilizar nombres de una sola palabra para denominar estas subrutinas es ayudar a planear el programa sin tener que preocuparse, en esta fase, de detalles como dar un número de línea. Si lo pensamos detenidamente, quizá decidamos que incluso la parte principal del menú del programa debería tratarse como una subrutina con el fin de mantener la sección principal del programa como un módulo independiente. La siguiente etapa para perfilar el programa constará de los siguientes pasos:

```
PROGRAMA PRINCIPAL (ENTRADA DE DATOS)
MENU (LLAMADA A SUBRUTINAS)
END
**SUBRUTINAS**
1 MENU
2 TOTALES
3 PROMEDIOS
(2) TOTALES
4 TOTALMES
5 TOTALCAT
(3) PROMEDIOS
6 PROMMES
7 PROMCAT
```

Este esquema de programa muestra que la subrutina de MENU ofrece una elección entre TOTALES y PROMEDIOS. Ambas serán subrutinas. La subrutina de TOTALES ofrece una nueva elección entre TOTALMES y TOTALCAT. Éstas serán las subrutinas que efectuarán los cálculos reales.

La subrutina de PROMEDIOS permite elegir entre PROMMES y PROMCAT, las cuales, asimismo, constituyen las subrutinas que efectuarán los cálculos adecuados. En este punto, debe ser posible ver si el "programa" hará lo que se le pide, sin desarrollar ningún código real (programa detallado escrito en BASIC). Si hasta aquí se han realizado todas las etapas correctamente, ya se está en condiciones de abordar la escritura de los módulos (subrutinas). El único cambio que debe hacerse en el programa principal será una llamada de subrutina antes de END, es decir, habría que añadir:

```
145 GOSUB **MENU**
```

Obsérvese que se continúa empleando "nombres" para designar las subrutinas, en vez de asignarles números de línea. Numerosos lenguajes, el PASCAL, por ejemplo, permiten designar con nombres a los subprogramas, pero, en cambio, ello no es posible en la mayoría de las versiones de BASIC, que, por el contrario, necesitan utilizar números de línea. Sin embargo, tomaremos en consideración estos detalles más adelante.

Veamos cómo puede escribirse la subrutina de MENU (se han omitido los números de línea pero el usuario puede añadirlos si desea realizar este programa).

```
REM SUBRUTINA **MENU**
PRINT "¿DESEA T(OTALES) O P(ROMEDIOS)?"
PRINT "DIGITAR T O P"
INPUT L$
IF L$ = "T" THEN GOSUB *TOTALES*
IF L$ = "P" THEN GOSUB *PROMEDIOS*
RETURN
```

Nota: Las llamadas a subrutinas se están acotando entre asteriscos: *— —*. Al programar, en vez de ello, se deberán emplear números de línea, que podrán incorporarse cuando se esté en condiciones de conocer cuáles son.

Imaginemos que se teclea T, inicial de TOTALES. Entonces el programa pedirá la subrutina de TOTALES. Ésta presentará otro menú, semejante al siguiente:

```
REM SUBRUTINA DE **TOTALES**
PRINT "¿DESEA TOTALES DE?"
PRINT "¿M(ES) C C(ATEGORIA)?"
PRINT "DIGITAR M O C"
INPUT L$
IF L$ = "M" THEN GOSUB *TOTALMES*
IF L$ = "C" THEN GOSUB *TOTALCAT*
RETURN
```

Supongamos que se elige M, indicativo de TOTALMES. Veamos cómo se puede escribir un módulo para calcular los gastos totales para cualquiera de los meses del año.

```
REM SUBRUTINA DE **TOTALMES**
REM CALCULA LOS GASTOS TOTALES DE
REM CUALQUIER MES
PRINT "ELEGIR MES"
PRINT "1-ENE 2-FEB 3-MAR 4-ABR 5-MAYO"
PRINT "6-JUN 7-JUL 8-AGO 9-SEPT"
PRINT "10-OCT 11-NOV 12-DIC"
PRINT "DIGITAR EL NUMERO DEL MES"
LET T = 0
INPUT M
FOR C = 1 TO 5
LET T = T + A(M,C)
NEXT C
PRINT "LOS GASTOS TOTALES DEL MES"
PRINT "NUMERO";M;"SON";T
RETURN
```

Se digita el número que representa el mes, y la orden INPUT asigna el número a la variable M(MES). M se emplea para especificar el subíndice de la "fila" de la matriz bidimensional A. El bucle FOR-NEXT incrementa el valor de C (columna) desde uno hasta cinco y, por consiguiente, la primera vez que se pasa el bucle, si se había elegido tres, es decir, el mes de marzo, la orden LET será equivalente a LET T = T + A(3,1). La próxima vez que se utilice el bucle, será equivalente a LET T = T + A(3,2), y así sucesivamente.

En este capítulo le encomendamos a usted la tarea de escribir las otras subrutinas o tratar de solucionar los otros ejercicios. Las matrices bidimensionales son ideales para cualquier programa que contenga tablas

Respuestas a los "Ejercicios" de la página 175

Función RND

```
40 IF R > 6 THEN LET R = 1
```

Bucle y promedio

```
5 FOR L = 1 TO 100
:
:
80 LET T = T + R
90 NEXT L
100 LET A = T/100
110 END
```

Sustitución por una subrutina

Suprima las líneas 5, 80, 90, 100 y 110 en la solución anterior. Cambie las líneas 10 a 70 por (pongamos por caso) las 1000 a 1070. Verifique que la línea 40 es igual que en la función RND de la solución anterior. Luego añada 1080 RETURN. Incorpore el resultado en el programa principal. Cambie las líneas 50 y 130 para que se lea 50 GOSUB 1000 y 130 GOSUB 1000.

INKEY$

```
10 PRINT "PULSE CUALQUIER TECLA"
20 LET A$ = INKEY$
30 IF A$ = " " THEN GOTO 20
40 PRINT "LA TECLA QUE USTED HA PULSADO ES";A$
50 END
```

Bucle de tiempos

```
5 PRINT "PULSE LA BARRA ESPACIADORA DESPUES DE
   10 SEGUNDOS"
10 FOR L = 0 TO 1
20 LET R = R + 1
30 IF INKEY$ = " " THEN GOTO 60
40 LET L = 0
50 NEXT L
60 PRINT "EL VALOR DE R DESPUES DE
   10 SEGUNDOS ES";R
70 END
```

Comparaciones IF... THEN

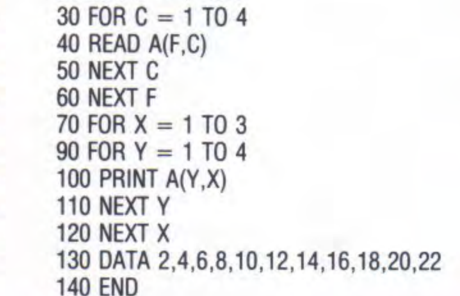
```
10 GOSUB 1000
20 PRINT "ADIVINE EL NUMERO"
30 FOR G = 1 TO 50
40 INPUT N
50 IF N > R THEN GOTO 110
60 IF N < R THEN GOTO 130
70 IF N = R THEN GOTO 150
80 NEXT G
90 PRINT "YA NO TIENE MAS OPORTUNIDADES.
   ¡HA PERDIDO!"
100 GOTO 500
110 PRINT "SU ELECCION ES DEMASIADO
   ELEVADA"
120 GOTO 80
130 PRINT "SU ELECCION ES DEMASIADO BAJA"
140 GOTO 80
150 PRINT "HA ACERTADO, FELICITACIONES"
500 END
1000 REM **SUBRUTINA ALEATORIA**
     (Introduzca aquí su subrutina)
1020 RETURN
```

de datos, ya sean éstos estadísticos, financieros o de cualquier otro tipo.

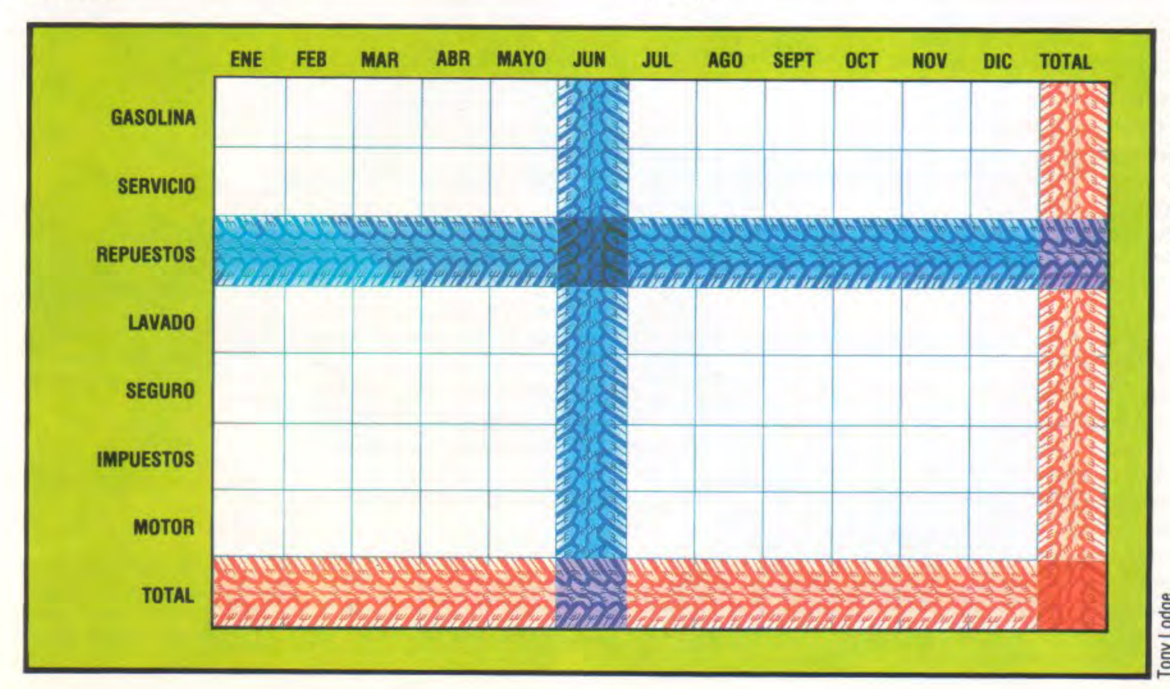
Ejercicios

■ **Asignación de valores** Escriba un programa que asigne valores a los elementos ("gasolina", "servicio", etc.) de la matriz representada en la parte inferior de esta página. A continuación, escriba una subrutina que requiera un mes y uno de los elementos e imprima el contenido del cuadrado determinado por ambos. Por último, escriba una subrutina que calcule el total de cada columna y sitúe el resultado al pie del cuadrado, haga lo mismo con cada fila y calcule el total final, que deberá colocar en el cuadrado inferior derecho.

■ **Bugs** El programa siguiente no funcionaría adecuadamente y produciría un mensaje erróneo. Descubrir el error y efectuar la corrección pertinente.



```
• 10 DIM A(3,4)
  20 FOR F = 1 TO 3
  30 FOR C = 1 TO 4
  40 READ A(F,C)
  50 NEXT C
  60 NEXT F
  70 FOR X = 1 TO 3
  90 FOR Y = 1 TO 4
  100 PRINT A(Y,X)
  110 NEXT Y
  120 NEXT X
  130 DATA 2,4,6,8,10,12,14,16,18,20,22
  140 END
```



Gastos de un coche
La figura muestra una cuadrícula de 8 × 13 cuadrados. Las filas representan elementos diferentes del coste de mantenimiento de un coche, y las columnas los diferentes meses del año. Realizar el ejercicio "Asignación de valores" para calcular el coste anual de mantenimiento de un coche

Tony Lodge

Trazadores de gráficos

Mediante estos dispositivos, un ordenador puede producir gráficos de alta calidad. Funcionan con plumas de punta de fibra y algunos de ellos pueden cambiar automáticamente de color

La capacidad de crear copias impresas de los diagramas que aparecen en la pantalla de un ordenador es un requisito esencial para muchas personas que utilizan la máquina profesionalmente. Ingenieros, científicos, diseñadores y hombres de negocios necesitan diagramas y cuadros con un grado de precisión que no pueden suministrar las impresoras convencionales. El único dispositivo que puede crear esas imágenes es un trazador de imágenes, y, hasta hace poco tiempo, éste resultaba excesivamente caro para el usuario del ordenador personal.

Sin embargo, con la introducción de dispositivos como el trazador de gráficos-impresora de cuatro plumas utilizado por el Tandy/CGP-115 y el Oric MCP-40, un terminal de gráficos puede estar al alcance de una gran parte de los usuarios. Recientemente ha aparecido en el mercado una gama completa de trazadores de gráficos con unas características que antes sólo podían ofrecer al usuario máquinas cuyo precio era prohibitivo.

La necesidad de utilizar un trazador de gráficos, o *plotter*, generalmente está determinada por el tipo de trabajo a que está destinado el ordenador. Un ingeniero o un proyectista necesitarán dibujos precisos de equipos y montajes, en cambio un hombre de negocios deseará cuadros y gráficos que muestren los volúmenes de ventas. Realizar esto con impresoras convencionales es un proceso muy laborioso y los resultados sólo aparecerán en blanco y negro. La otra única opción de bajo coste es efectuar una fotografía de la pantalla, pero si bien esto puede ser suficiente para gráficos comerciales, no reúne los requisitos mínimos de precisión que pedirían un arquitecto o un diseñador.

Los trazadores de gráficos funcionan de una forma completamente distinta a las impresoras: trazan líneas entre dos puntos en lugar de partir de formas preestablecidas o modelos de puntos.

El principio básico con el que funcionan todas las marcas consiste en un sistema de coordenadas X, Y. Al igual que una gráfica puede ser trazada definiendo las coordenadas por las que debe pasar la línea, también una figura puede ser descompuesta en una serie de coordenadas. Para poder unir estas coordenadas con el fin de recrear la figura, tiene que existir alguna forma de movimiento. Por ello se fija la pluma a un pantógrafo que puede desplazarse en el sentido de las abscisas X (de izquierda a derecha) al mismo tiempo que la pluma se mueve a lo largo del pantógrafo en el sentido de las ordenadas Y (de arriba abajo).

El tipo tradicional de trazador de gráficos se conoce con el nombre de *lecho plano*, debido a que el papel es fijado a una placa plana, sobre la cual se desplaza el pantógrafo, tal como se puede apreciar en la ilustración. Esto tiene el inconveniente de que el trazador debe ser, como mínimo, tan grande como la hoja de papel.

Soporte pluma
Se pueden cambiar automáticamente hasta tres plumas. El pantógrafo regresa al soporte y cambia la pluma en uso por el nuevo color necesario. Manualmente se pueden cambiar más colores

Clips magnéticos
Mantienen el papel plano sobre el trazador. Están hechos de un material magnético flexible

Portaplumas
La pluma que se está utilizando se fija (en este caso magnéticamente) en el soporte, que se desplaza hacia abajo y sitúa la pluma en contacto con el papel

Un método para reducir su tamaño es adoptar una versión a gran escala del modelo del trazador de cuatro plumas (véase ilustración), en el cual el papel se desplaza en una dirección y las plumas en la contraria. A este tipo pertenecen los de las marcas Strobe 100 y Hewlett Packard Sweetlips. El movimiento del papel debe ser controlado con la misma precisión que el desplazamiento del pantógrafo en los del tipo de lecho plano, y esto se logra con un motor paso a paso. Este tipo de motor posee unas características muy especiales, y cada vez que recibe un pulso de la potencia aplicada, realiza una fracción de giro. Se aplica principalmente a unidades de disco, en las cuales controla la posición del cabezal sobre la superficie del disco, y en robótica (véase p. 176).

La conexión de un trazador a un ordenador es, por lo general, similar a conectar una impresora, por lo menos en lo que se refiere a la interface. Normalmente, pueden adquirirse trazadores con interfaces en serie (RS232) o en paralelo (Centronics o IEEE488),

que pueden enchufarse a la conexión utilizada por las impresoras. A menudo, su programación es algo más complicada. En vez de sólo enviarle los resultados de un programa que debe ser impreso, ahora será necesario suministrarle también información sobre cómo deben ser presentados. Por lo común, esto se realiza de una forma muy parecida a la empleada para formar un diagrama en la pantalla.

Debido al complicado sistema que utilizan los trazadores para realizar su cometido, éstos son, por regla general, "inteligentes". Esto quiere decir que poseen microprocesadores que convierten los caracteres e instrucciones recibidos del ordenador en una serie de coordenadas, que luego dibuja el trazador. Muchos de los más perfeccionados permiten también dibujar figuras complejas, tales como círculos y curvas, proporcionándoles únicamente los puntos de partida: el trazador hará el resto. El rotulado de los gráficos y diagramas y el coloreado de diagramas de barras y de segmentos circulares son, frecuentemente, procesos automáticos, factor que hace la programación mucho más sencilla.

Muchos trazadores se suministran junto con el software, lo cual les permite estar conectados directamente al interior del programa, en vez de ser como una copia sobre papel de la pantalla. Si no se dispone de este tipo de programas, el usuario debe desarrollar las rutinas necesarias para transformar la información de la pantalla en los códigos adecuados para guiar al trazador. Algunos de ellos no disponen de juegos de caracteres y, por consiguiente, deben crearse incluso los códigos para las letras y números. En cambio, esto permite al usuario crear sus propios caracteres y tipos.

Una vez que se ha generado una figura, ésta puede trazarse en cualquier posición, orientación o tamaño, con lo cual puede obtenerse una colección de formas para diferentes usos. Con frecuencia son muy útiles las rutinas para trazar círculos, curvas y figuras en secciones de gráficas, especialmente en el campo de los gráficos comerciales. Sin embargo, los principios de creación de un dibujo a partir de las coordenadas de la pantalla son exactamente los mismos que para crear la figura sobre el papel y, por lo tanto, la programación es bastante sencilla.

Pantógrafo
Puede situarse en cualquier punto a lo largo de la hoja (eje de las X) y el portaplumas es desplazado según el eje de las Y. La combinación de los movimientos de izquierda a derecha y de atrás adelante permite llegar a cualquier punto de la página

Motores paso a paso
Estos motores giran sólo unos grados por cada pulso eléctrico. Accionados adecuadamente, proporcionan el movimiento preciso de la pluma y el pantógrafo

Tablero de circuitos
Normalmente, los plotters son dispositivos "inteligentes" (pueden dárseles órdenes de alto nivel, por ejemplo: "dibujar una circunferencia de radio y centro determinados", y el trazador mueve la pluma para dibujar ésta). El tablero de circuitos dispone de su propio microprocesador, ROM y RAM

Conexión interface
Los trazadores se conectan al ordenador mediante una interface estándar, por ejemplo la RS232 (en serie) o Centronics (en paralelo). Para el ordenador significa lo mismo que una impresora, aunque para su funcionamiento serán necesarias órdenes distintas

Control elevador de la pluma
Permite situar manualmente la pluma en contacto con el papel, o bien sin que toque a éste

Controles de la pluma
Mediante estos mandos, la pluma puede desplazarse manualmente a cualquier punto del papel

ERROR

David Weeks

Trazador de gráficos-impresora de cuatro plumas

Este mecanismo atrajo la atención de la industria cuando apareció por primera vez en la impresora Sharp CE-150. Sus hermanos mayores: el Tandy's CGP-115 y el Oric MCP-40, ayudaron a que el coste de las impresoras en color fuera asequible para el usuario del ordenador doméstico.
Como todas las buenas ideas, el sistema es conceptualmente muy simple. Un rollo de papel es empujado a través del mecanismo por medio de un piñón. El papel es desplazado, hacia adelante y hacia atrás, en intervalos muy precisos, mientras un portaplumas que sujeta los cursores de los cuatro colores se desplaza por la superficie, de izquierda a derecha y viceversa.
El portaplumas, para crear el texto o el gráfico, gira hasta enfrentar el color correcto y luego la pluma es presionada contra el papel. Las líneas horizontales se obtienen por el desplazamiento de la pluma y con el papel inmóvil; las verticales, con el papel en movimiento y la pluma estática. Las combinaciones de ambos movimientos producen diagonales o curvas. La calidad de la impresión es notable, pero la limitada anchura del papel la hace inadecuada para procesamiento de textos u otros fines profesionales

Chris Stevens

Alan Turing

El test para determinar la inteligencia de una máquina tomó su nombre de este matemático británico. La mayor parte de su trabajo, sin embargo, lo realizó para el servicio de inteligencia militar durante la última guerra mundial

Ciencia y deporte
Alan Turing (1912-1954) se inspiraba y relajaba mediante las carreras de fondo. Estaba intrigado por el efecto del esfuerzo físico sobre la creatividad y agilidad mental



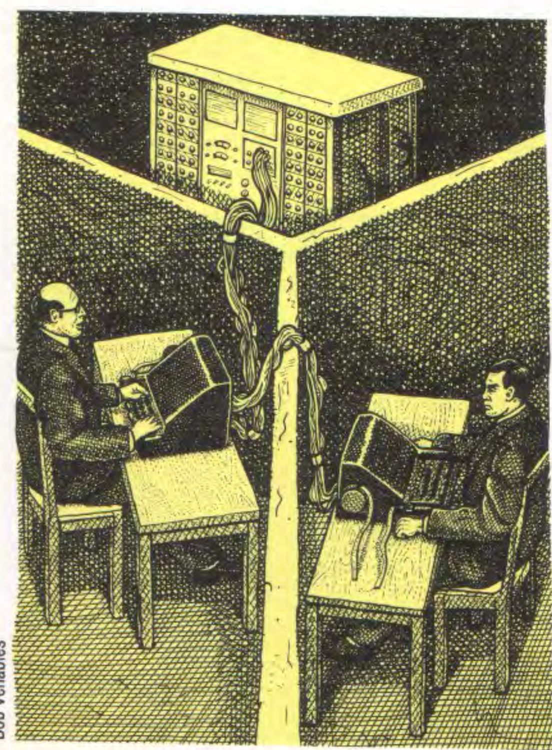
¿Pueden pensar las máquinas?
Para contestar a esta pregunta, Turing propuso su famoso test llamado *juego de la imaginación*, y conocido también como *test de Turing*. Una persona se sitúa en una habitación con una teleimpresora, la cual está conectada a otra teleimpresora, colocada en una habitación diferente y manejada también por un operador, así como al ordenador que se quiere probar. La primera persona puede hacer cualquier pregunta que desee a uno de los dos. Si, consecuentemente, ésta es incapaz de determinar cuándo está en comunicación con la otra persona y cuándo con el ordenador, la máquina puede ser considerada como inteligente. Después de todo, continúa la tesis, no tenemos forma de asegurar si otras personas razonan o son conscientes, excepto comparando sus reacciones con las nuestras propias frente a un hecho determinado

El joven Alan Turing mostró una extraordinaria intuición científica. Estando en la escuela, escribió a su madre: "Me parece que siempre quiero hacer las cosas a partir de aquello que abunda más en la naturaleza". Los matemáticos ofrecen pronto muestras de su talento, y Turing, tan pronto supo leer y escribir, empezó a componer versos y a diseñar bicicletas anfibias.

Mientras su padre se hallaba en Madrás, trabajando para el Indian Civil Service, Turing ganó numerosos premios escolares y más tarde una beca que le llevaría al King's College de Cambridge. Durante su permanencia en esta ciudad, primero como estudiante y luego como miembro de la junta de gobierno de su colegio, fue cuando su interés empezó a centrarse en los problemas de la lógica matemática.

En 1931, el matemático checo Kurt Gödel asombró al mundo científico con el descubrimiento de que había teoremas matemáticos que eran verdaderos aunque nunca pudieran probarse. Alan Turing se puso a investigar aquellos que podían ser probados.

Propuso una máquina, cuya construcción dejó a la imaginación, que podría realizar de manera mecánica los procesos que por regla general lleva a cabo un matemático. Para cada proceso había una máquina: por ejemplo, una máquina para sumar, otra para dividir, y una tercera para integrar, y así sucesivamente. Más tarde estas máquinas serían conocidas como máquinas Turing.


Bob Venables

El joven matemático investigó los trabajos de estas máquinas imaginarias y llegó a una notable conclusión. En vez de destinar la realización de cada proceso matemático a una máquina independiente, era posible diseñar un aparato "universal" que fuera capaz, al ser "programado", de cumplir las funciones de cualquiera de las máquinas especializadas. Alan Turing había encontrado así, casualmente, la teoría del ordenador programable.

Al estallar la segunda guerra mundial, Turing fue rápidamente alejado del mundo académico y reclutado por la Escuela de Códigos y Cifrados del gobierno británico, situada en Bletchley Park (Buckinghamshire). Si no hubiera sido por el conflicto bélico, quizá sus máquinas hubieran continuado siendo una entelequia.

Las actividades que se realizaban en Bletchley Park, que eran altamente secretas, consistían de manera primordial en descifrar con suma urgencia los códigos militares alemanes.

Debido a que estos códigos podían ser cambiados diariamente, era necesario el empleo de máquinas para su desciframiento antes de que los servicios alemanes introdujeran nuevos códigos. Bletchley Park se convirtió en un inmenso centro de procesamiento de información. Promediada la guerra, Turing fue enviado a Estados Unidos, con el fin de que creara unos códigos seguros para las comunicaciones trasatlánticas entre los países aliados.

Debido al carácter secreto de su trabajo en esta época, se dispone de muy pocos datos sobre su actividad. Sin embargo, se cree fundamentalmente que conoció a Von Neumann cuando se hallaba en Princeton (Nueva Jersey). Hacia el final de la guerra, Turing recibió el encargo de empezar los trabajos para la construcción de un ordenador totalmente británico, destinado al National Physical Laboratory, y que recibiría el nombre de ACE (Automatic Computing Engine).

Esta máquina tardó mucho tiempo en ser construida, pero muchas de sus características eran superiores a las del ENIAC (véase p. 46). Frustrado por el lento avance, Turing dimitió y se fue a vivir a Manchester, donde empezó a colaborar en el proyecto del ordenador de la universidad. Al mismo tiempo, era asesor de la compañía Ferranti y, por tanto, colaboró en la construcción de los primeros ordenadores fabricados en Gran Bretaña.

Turing fue una personalidad excéntrica, que se dedicó a lo que sabía que era importante sin preocuparse de prejuicios sociales o cortapisas legales. Un amigo dijo de él que era "divinamente retrasado" para ver los errores de los otros, pero su genio científico era intachable. En 1952 fue condenado por acusaciones relativas a homosexualidad y, dos años más tarde, se suicidó. ¿Quién puede predecir cuál hubiera sido su contribución al desarrollo de la inteligencia artificial si no hubiera adoptado esa trágica determinación?

Vuelos de fantasía

Como el funcionamiento de un complejo avión comercial o militar se puede simular a la perfección, hoy los pilotos efectúan en tierra la mayor parte de su adiestramiento



Preparado para el despegue
"Bravo Alfa tres dos cinco, está autorizado para despegar desde la pista uno dos. La velocidad del viento es de unos cinco nudos, en dirección uno cero cinco. Llámeme cuando haya recorrido mil metros."
Pero la pista que se extiende ante este Boeing 737 existe sólo como impulsos digitales en el interior del sistema de simulación de vuelo por ordenador Novoview SP3 Rediffusion

Ya en la década de los cuarenta los pilotos de aeroplanos recibían parte de su entrenamiento inicial no en aeronaves reales sino en modelos estilizados a escala natural denominados Link Trainers. Estas máquinas, que representaron la primera generación de simuladores de vuelo, eran bastante rudimentarias y servían sólo para dar una ligera idea de los efectos de los mandos del avión, con la finalidad de enseñar las maniobras básicas de vuelo.

Con la introducción de la primera generación de aviones de transporte a reacción de motores múltiples, como el Comet de Havilland, se hizo patente que, para la enseñanza y práctica de vuelo en distintas clases de aviones, se necesitaba un método más seguro y económico, que prescindiera del adiestramiento real en el espacio aéreo. Los fabricantes de electrónica respondieron produciendo los primeros simuladores controlados por ordenador, que utilizaban técnicas tanto analógicas como digitales.

Montados sobre arietes hidráulicos, estos simuladores eran una réplica de las secciones de la cabina de la nave en la que los alumnos aprendían a volar. El control por ordenador de los sistemas hidráulicos permitía que la maqueta respondiera al movimiento de los mandos de vuelo tal como lo haría la aeronave verdadera, y también permitía que el instructor introdujera situaciones "de emergencia".

El interior de la cabina era una réplica exacta de la verdadera, con todos los instrumentos y el equipo de mandos. Estos últimos no estaban diseñados sólo para el piloto, sino para toda la tripulación, puesto que para entonces ya era de importancia vital que la totalidad de los miembros de la tripulación del aparato trabajara en equipo.

El único elemento que faltaba era la simulación de la vista que tendría el piloto desde el parabrisas durante el despegue y el aterrizaje. Los primeros intentos que se efectuaron en este sentido se centraban en la proyección de una película, pero las dificultades que suponía la constante reorientación hacían que esta solución distara mucho de ser perfecta.

Como suele suceder tan a menudo, en el momento en que el problema se hacía realmente acuciante,



Cortesía de Rediffusion Simulation Ltd.

Aviones con patas
Vistos desde fuera, los simuladores de vuelo no parecen más que extrañas estructuras cuadrangulares apoyadas en soportes hidráulicos angulares. Sin embargo, al penetrar en su interior uno se encuentra con un mundo totalmente diferente: la cabina de mando de una moderna aeronave

desde otra rama de la tecnología se proponía la auténtica solución. Hemos estudiado cómo las imágenes generadas por ordenador han revolucionado la producción de películas de dibujos animados (véase p. 181). Estas mismas técnicas se utilizan en la actualidad en los simuladores de vuelo para proporcionarles a los pilotos imágenes en movimiento que muestren el aspecto que ofrece un aeropuerto a medida que el avión se aproxima a él. Y, dado que la imagen animada se retiene en software y se puede cargar desde un disco, un sistema de simulador le puede proporcionar al piloto un gran repertorio de aeropuertos.

Un empleo aún más trascendente de esta técnica es el de los simuladores de vuelo que utilizan los pilotos militares. En este caso el objetivo no es sólo el de adaptar a los pilotos a uno u otro tipo de aeronave, ya que los sistemas de aviónica modernos son tan perfectos que los aviones prácticamente vuelan por sí mismos. Los sistemas de adiestramiento de pilotos y observadores con fines militares se orientan más a las técnicas de reconocimiento de objetivos y para apuntar las armas, tanto de aire-tierra como de aire-aire.

Al mando
El software del simulador de vuelo Microsoft coloca al usuario al mando de un avión Cessna 182 de un solo motor, que emprende el vuelo desde la réplica de uno entre veinte campos de aviación verdaderos de los Estados Unidos. Una réplica muy completa del panel de instrumentos ocupa la mitad inferior de la pantalla del monitor, mientras que la mitad superior representa una panorámica en perspectiva del "mundo exterior". El paquete, que incluye un juego de combate entre aviones de caza, es más bien un auténtico programa de aprendizaje auxiliado por ordenador, ideado para enseñar las pautas básicas del vuelo

No obstante, la necesidad básica sigue siendo la de presentarle a la tripulación de vuelo una panorámica desde la cabina que sea lo más parecida posible a la realidad.

Cuando hablábamos de los programas de hoja electrónica (véase p. 158), describimos una técnica denominada "de ventana" que le permitía al operador, por así decirlo, pasar la pantalla del monitor por una hoja más grande que la que aquél podía contener. Una técnica similar es esencial para el funcionamiento de un simulador de vuelo avanzado, en el que la cabina se mueve realmente como lo haría el avión verdadero en respuesta a los mandos y en relación con el "escenario" exterior. De modo que cuando, por ejemplo, el piloto determina un giro en picado, el aspecto que ofrece el terreno que se presenta ante él cambia en consecuencia.

Proyecciones aéreas
Se utilizan tres proyectores de televisión (rojo, azul y verde) para reflejar las imágenes animadas del mundo exterior sobre una pantalla situada frente al piloto, mediante un espejo corrector que se ajusta para compensar la distorsión producida por la curvatura hacia atrás de la pantalla

En un simulacro de combate aire-aire la situación se complica aún más debido a las maniobras que realiza al enemigo, y éstas han de escogerse aleatoriamente entre un gran archivo de posibles maniobras, para que al piloto se le presente una situación a la cual él pueda responder con flexibilidad, adoptando una decisión libre y nueva en cada ocasión, prescindiendo de técnicas aprendidas de memoria.

Las escuelas de adiestramiento militar llegan aún más lejos con sus "entrenamientos por simulación". Utilizan imágenes de alta resolución, fotografías por satélite enviadas desde aviones de reconocimiento en vuelos a gran altura, e informes de los agentes para producir una película animada que muestre con sumo detalle las características físicas del terreno sobre el cual podría volar un piloto en una misión de bombardeo a poca altura.

Utilizando esta película junto con un simulador de vuelo se obtiene una perfecta y total recreación de una misión. Este método de adiestramiento permite que el piloto realice todo el trayecto completo de su incursión, desde que despega hasta que regresa a aterrizar en su base, sin que para ello haya de moverse de tierra en ningún momento. La misma flexibilidad de la generación de imagen que posibilita que los pilotos comerciales "practiquen" despegues y aterrizajes en una variedad de aeropuertos, le permite al piloto militar experimentar una amplia variedad de misiones y objetivos.



Luego de haberse desarrollado como un medio auxiliar de adiestramiento para miembros de tripulaciones de vuelo, las técnicas de simulación están siendo utilizadas en la actualidad por la marina mercante para familiarizar a los oficiales de guardia con el aspecto de la aproximación a puertos y canales; por los buzos para el reconocimiento de objetos en condiciones de poca visibilidad, y por los astronautas para reproducir el ambiente interespacial. El potencial de las técnicas para generar imágenes, en actuación conjunta con los usuarios y bajo control por ordenador, hace posible recrear virtualmente cualquier experiencia visual, lo cual, a su vez, ha dado lugar a que recientemente apareciera en el mercado toda una nueva generación de juegos y entretenimientos.

Los programas para juegos que actúan como simples simuladores de vuelo constituyen un claro ejemplo de este fenómeno, al desafiar al usuario a realizar un vuelo exitoso, bajo una serie de condiciones adversas: poco combustible, escasa visibilidad, un fallo del motor, etc. Pero no por ello debe uno pensar que estos juegos estén directamente relacionados con los sistemas profesionales que llevan el mismo nombre, con la única salvedad, quizá, que supone el caso del simulador de vuelo Microsoft.

Cualquier juego por ordenador, sin embargo (incluso el ya obsoleto "Ping-pong", precursor de todos ellos), que represente en el monitor de televisión de un ordenador una situación "de la vida real", se puede considerar como un simulacro. La única diferencia real reside en la complejidad de la representación y, tal vez, en la calidad de la imagen generada por ordenador.

Estos simulacros pueden ser sólo un producto de la imaginación, como "invasores del espacio" (*Space invaders*) y Pac-Man. En esos casos el diseñador elaboró los parámetros del juego y los tradujo en posibles cursos de acción. O pueden ser simulacros más directos de la realidad, como los diversos juegos de carreras de coches.

No todo el software de simuladores de vuelo basado en microordenadores personales está pensado para quienes gustan de los juegos. Existen paquetes importantes de aprendizaje auxiliado por ordenador en el campo de la navegación aérea, el control del tráfico aéreo y la planificación de vuelo, por ejemplo. No obstante, el simulador de vuelo Microsoft, para el ordenador personal IBM, es a la vez un juego recreativo y un excelente ejercicio para futuros pilotos.

En la primera parte del manual se explica que el paquete reproduce todos los instrumentos y equipos necesarios, según el reglamento de la US Federal Aviation Authority (Dirección Federal de Aviación de Estados Unidos), para el vuelo, tanto diurno como nocturno, por instrumentos o a través de la vista. La



mitad inferior de la pantalla representa el panel de instrumentos de un Cessna 182 (el avión de un solo motor sobre el cual se basa el simulacro), mientras que la mitad superior queda reservada para imágenes animadas en color de resolución media relativas a la panorámica que se observa desde el parabrisas. Las imágenes incorporan efectos seudotridimensionales, es decir, que se percibe una sensación bastante real de movimiento a partir simplemente de la variación de las perspectivas.

El simulador de vuelo incluye su propio mapa del terrreno, que cubre la superficie de Estados Unidos y



Cortesía de Rediffusion Simulation Ltd.

«Avión enemigo... en dirección 030A...»
Las imágenes de esta calidad generadas por ordenador permiten que tanto las tripulaciones de aire como las de tierra realicen sus prácticas, como las tareas de reconocimiento, en cualquiera de las condiciones meteorológicas o de visibilidad que escoja el instructor. Esta imagen, por ejemplo, producida con el sistema Novoview Rediffusion Simulations, nos muestra dos aviones norteamericanos A10 de ataque al suelo

parte de Canadá, México y el Caribe, abarcando unos veinte campos de aviación diferentes de cuatro zonas principales: Nueva York-Boston, Chicago, Los Ángeles y Seattle.

Teóricamente se puede "volar" atravesando en línea recta todo el país, desde Nueva York a Los Ángeles vía Chicago, pero como, excepto en estas populosas áreas, no existen puntos de reabastecimiento, el "piloto" se estrellaría muy pronto porque se le acabaría el combustible. Incluso aunque fuera posible "llevar" suficiente combustible, el simulador trabaja en tiempo real, lo que significa que, a una velocidad máxima de aproximadamente 240 km/h, el "viaje" duraría alrededor de 20 horas, tiempo que en su mayor parte sería de vuelo recto y nivelado a campo abierto. El paquete, por lo tanto, permite al usuario "saltar" de una zona a otra y ofrece la posibilidad de decidir la hora del día, la estación (que determina a qué hora amanece y anochece), las condiciones meteorológicas, incluyendo grado de nubosidad, velocidad y dirección del viento y un factor para la turbulencia del aire.

El simulador de vuelo Microsoft se basa en el tipo de avión más sencillo de todos. A pesar de ello, el programa se ha podido comprimir en 64 Kbytes gracias a la utilización de un código de lenguaje máquina de altísima eficacia. Reproducir la cabina de mandos completa de un moderno jet de cuerpo ancho, permitiendo que la réplica se mueva verdaderamente en respuesta al accionamiento de aquéllos, es una tarea muchísimo más compleja.



Información clasificada

Al diseñar cualquier programa que suponga almacenamiento o manipulación de información, es importante atender a la estructura, de modo que no sea preciso duplicar ningún dato

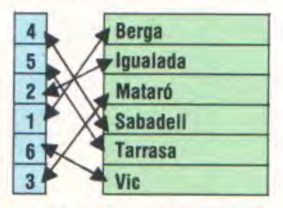
El camino más corto
Una empresa tiene intereses en seis poblaciones y todos los meses ha de enviar un camión a recorrerlas. El ordenador de la empresa puede elaborar la ruta más conveniente.

Ciudad$(1)	"Berga"
Ciudad$(2)	"Igualada"
Ciudad$(3)	"Mataró"
Ciudad$(4)	"Sabadell"
Ciudad$(5)	"Tarrasa"
Ciudad$(6)	"Vic"

Los nombres de las ciudades están incluidos en un archivo secuencial en cinta por orden alfabético. Se leen por este orden en el archivo de la matriz Ciudad$( ). El ordenador calcula que el orden más adecuado es:

Sabadell	Ciudad$(4)
Tarrasa	Ciudad$(5)
Igualada	Ciudad$(2)
Berga	Ciudad$(1)
Vic	Ciudad$(6)
Mataró	Ciudad$(3)

En lugar de volver a almacenar los nombres de las ciudades por ese orden, utilizando más memoria, el ordenador almacena solamente los números de las ciudades en la matriz Ciudad$( ), de la siguiente manera:



La lista de números es un índice de la matriz Ciudad$( ). Se podrían crear otros índices de este tipo para la matriz, cada uno de los cuales ordenaría los nombres de distinta manera. La ventaja de confeccionar índices como éste es que no se debe clasificar ni duplicar el archivo original: sólo se necesita clasificar y almacenar los índices

Toda persona que haya consultado alguna vez un sistema de índice por fichas (como el archivo de una biblioteca, por ejemplo) sabe la gran utilidad que ofrece. Si a usted en alguna ocasión se le ha caído al suelo un fichero, sabrá también que esas mismas fichas, pero desordenadas, se convierten en objetos sin ninguna utilidad. Una biblioteca contiene una vasta cantidad de información, pero a menos que se distribuyan los volúmenes según algún tipo de orden, su valor como sistema de información es prácticamente nulo.

La esencia de un sistema de información no reside en la información en sí misma sino más bien en la forma en que ésta esté organizada. Tomemos, por ejemplo, este dato de una guía comercial:

Pérez, J., Calle F, Barcelona.

En sí mismo, el valor de esta información es limitado; sin embargo, si se sabe que proviene de la sección "Ferreterías" de la guía, adquiere un nuevo significado, determinado por la estructura en la cual se halla incluida.

La estructura más sencilla de información es la que se ofrece en forma de archivo: una serie de datos con ciertas características en común. El nombre del archivo revela algo acerca de la información que contiene, y poner toda la información junta bajo un mismo título hace que su empleo resulte mucho más sencillo. El archivo se puede tratar como una gran unidad de información o como una agrupación particular de unidades más pequeñas. Un libro es un archivo; una novela, por ejemplo, se lee normalmente como una totalidad, mientras que un libro de cocina por lo general se lee como un conjunto de recetas individuales.

Si el archivo es grande, encontrar una información determinada puede suponer tener que comenzar por el primer dato del archivo e ir explorando cada dato hasta hallar el deseado. Esto se conoce como *búsqueda* o *acceso secuencial*, y el archivo organizado de este modo es un *archivo secuencial*. Un programa de televisión es un archivo secuencial de información y lo mismo se puede decir de una conversación entre dos personas.

Los archivos secuenciales son muy comunes porque su implementación es útil y económica y además, en muchos sentidos, reflejan los métodos del pensamiento humano. No obstante, su consulta puede resultar lenta y pesada, de modo que con frecuencia se dividen internamente en subarchivos que se pueden hallar de manera directa sin tener que buscar en todo el archivo. En otro tiempo, los libros eran simples archivos secuenciales, pero sufrieron una transformación a raíz del invento de los capítulos, los números de página y el índice. Los capítulos son los subarchivos del libro, y las páginas son los subarchivos tanto de los capítulos como del libro.

Se denomina *archivo de acceso directo* a aquel que no requiere búsqueda secuencial. Un álbum de canciones en cinta magnética es un archivo secuencial, mientras que el mismo álbum pero en un disco de larga duración es un archivo de acceso directo: hallar una canción en la cinta requiere comenzar por el principio y bobinar hacia adelante, mientras que en el caso del disco se puede acceder directamente a cada surco moviendo el brazo del tocadiscos sobre el LP y colocándolo al comienzo del surco deseado.

El acceso directo depende de que uno sepa dónde se encuentra lo buscado: en un libro el índice puede señalar en qué página comienza un capítulo o se halla determinada ilustración. Saber con exactitud dónde están las cosas significa trabajo (y, por lo tanto, cuesta dinero). Confeccionar el índice de un libro representa un trabajo extra para el autor o el editor, y cabe en lo posible que la información que proporciona el libro no justifique ese gasto.

Los ordenadores procesan grandes cantidades de información a gran velocidad, utilizando una variedad de estructuras de datos. Éstos se han de almacenar de forma permanente en cinta magnética o en disco de algún modo estructurado (generalmente, en archivo secuencial), pero de manera muy diferente en la memoria principal del ordenador.

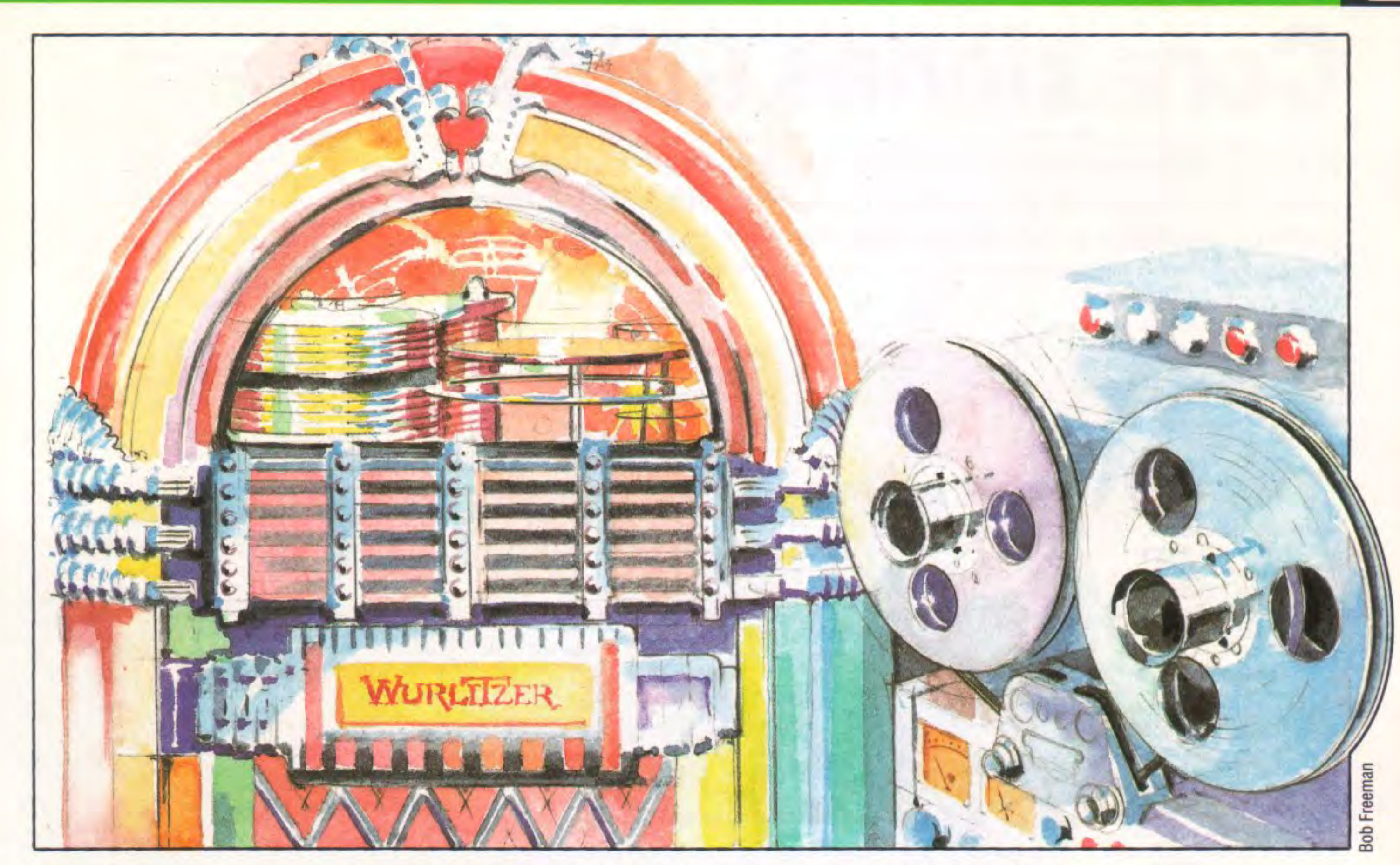
Supongamos que una distribuidora tiene las direcciones de todas las tiendas a las que vende sus productos en un archivo secuencial en cinta y que desea que el ordenador imprima las listas de entrega para que dispongan de ellas los conductores que llevan a cabo el reparto a las tiendas. El archivo en cinta podría ser entonces de la siguiente manera:

```
Casa Bermúdez       Calle C
González Hnos.      Calle D
Modas Rowena        Calle A
Montes de Gredos    Calle F
Roig y Pujol, S.A.  Calle B
Viuda de Zárate     Calle E
```

Cuando se lea el archivo de la cinta a la memoria principal, cada nombre y dirección se almacenará en una dirección numerada individual, y todas estas direcciones juntas constituirán un bloque de memoria con un nombre propio, de modo que en la memoria el archivo es así:

```
NOMBRE DEL BLOQUE: Tiendas
1) Casa Bermúdez       Calle C
2) González Hnos.      Calle D
3) Modas Rowena        Calle A
4) Montes de Gredos    Calle F
5) Roig y Pujol, S.A.  Calle B
6) Viuda de Zárate     Calle E
```

Ahora se puede acceder individualmente a cada uno de los datos refiriéndose al bloque y a su dirección dentro del mismo. Tiendas (4), por ejemplo, contiene Montes de Gredos, Calle F. Esta estructura se denomina matriz (véase p. 194) y es la estructura de datos utilizada más comúnmente por los ordenadores para el procesamiento interno de datos. Es como un libro



Bob Freeman

con una información determinada en cada página. Observe que esta sencilla estructura modifica de inmediato la forma en que observamos los datos. Un archivo de nombres y direcciones reconocibles se ha convertido en un bloque de datos anónimos. Los ordenadores no necesitan saber qué significa un dato determinado, sino sólo dónde está y qué deben hacer con él.

Los datos de la matriz Tiendas ( ) están clasificados alfabéticamente, pero es poco probable que este orden sea el más adecuado para visitar las tiendas. Supongamos que el ordenador determina que el mejor orden de entrega es:

1)	Montes de Gredos	Calle F
2)	Casa Bermúdez	Calle C
3)	Modas Rowena	Calle A
4)	González Hnos.	Calle D
5)	Viuda de Zárate	Calle E
6)	Roig y Pujol, S.A.	Calle B

Este plan de entrega se puede almacenar en otra matriz, pero ello implicaría almacenar dos veces en la memoria la misma información. Los propietarios de micros sabrán que la RAM es limitada, y puede ser inadecuado o bien imposible duplicar los datos de esa manera; de modo, entonces, que se necesita otro método.

Si los datos verdaderos se reemplazan por los números que ostentaban en la matriz Tiendas ( ), el plan de entrega sería el siguiente:

NOMBRE DEL BLOQUE: Entregas

1) 4
2) 1
3) 3
4) 2
5) 6
6) 5

Lo anterior significa en realidad: "Vaya primero a la tienda cuyas señas están almacenadas en Tiendas (4), vaya luego a Tiendas (1), después a Tiendas (3)...", y así sucesivamente. La única información significativa del plan es el orden en que los conductores que efectúan el reparto de la mercancía deben visitar las tiendas, de modo que eso es lo único que se ha de almacenar en la nueva matriz, Entregas.

Entregas ( ) es ahora un índice de la matriz Tiendas ( ) con el fin de determinar las entregas. Al imprimir este plan, el ordenador utilizará los números de la matriz Entregas ( ) para imprimir los nombres y las direcciones de la matriz Tiendas ( ) en el orden correcto.

En este sencillo ejercicio, la información (los nombres y direcciones de las tiendas) se ha manipulado pero no se ha modificado en razón de las distintas estructuras de datos que se le han impuesto. Una estructura de datos no modifica el contenido de éstos, sino que les otorga un significado al asociarlos de forma ordenada con otros datos.

Así como podemos reordenar la matriz Tiendas ( ), clasificándola esta vez de acuerdo a la matriz Entregas ( ), de la misma manera podemos confeccionar otros índices con otras finalidades. Cuando hablábamos en las bases de datos (véase p. 124), comentamos que cierta información se podía seleccionar tomando como referencia indicadores incluidos en cada registro individual. De este modo podemos "colocar" en cada registro del archivo Tiendas ( ) un indicador que señale su lugar en el plan de entregas. Podemos ampliar el registro aún más para incluir, por ejemplo, un indicador en un archivo de pedidos fijos; Casa Bermúdez, pongamos por caso, siempre encarga 50 camisas de manga larga, 15 de manga corta, etc. El departamento de producción podrá entonces extraer del archivo la información relativa sólo al número de camisas que necesita confeccionar.

En el surco correcto
Un juke-box contiene 200 canciones grabadas en 100 discos y pesa 80 kg. Para seleccionar una canción hay que pulsar tres teclas. El tiempo promedio entre SELECT y PLAY es de 15 segundos. En una grabadora de cinta, un carrete de cinta magnética puede contener las mismas 200 canciones y pesa sólo 10 kg. Para seleccionar cualquier canción hay que rebobinar la cinta, pulsar PLAY y esperar. Entre SELECT y PLAY median alrededor de 1 500 segundos.
Un juke-box es una máquina de acceso directo: es rápida, bastante especializada y cara. Por el contrario, una grabadora de cinta es un aparato de acceso secuencial: lento, mucho menos especializado que el juke-box, pero de precio más asequible. Una grabadora de cassette conectada a un microordenador es un dispositivo de acceso secuencial, mientras que una unidad de disco flexible es un dispositivo de acceso directo, aun cuando se pueda utilizar para almacenar archivos secuenciales

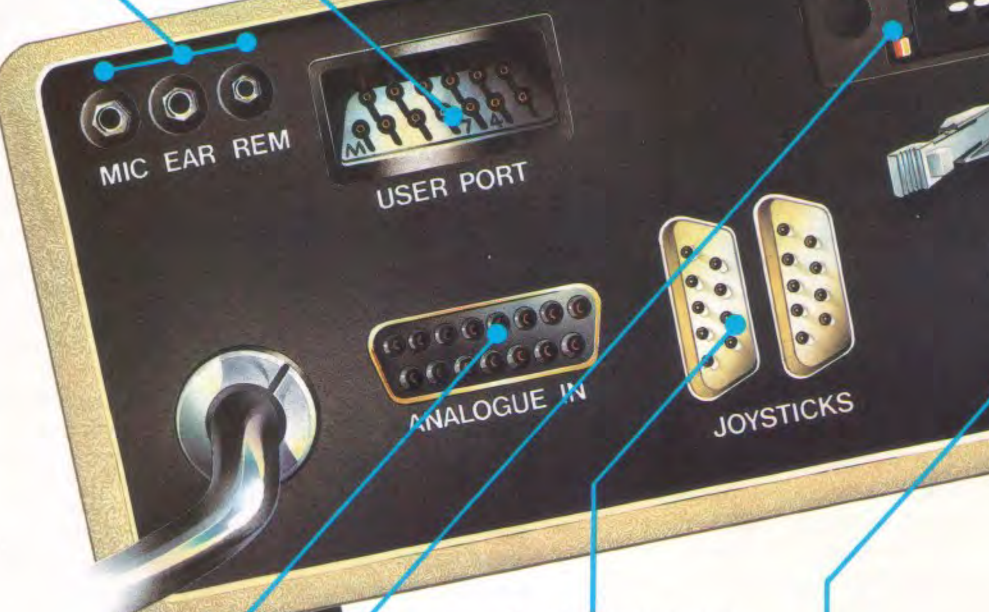
Conexiones útiles

No se puede enchufar una clavija cuadrada en un agujero redondo; tampoco es posible conectar entre sí dos dispositivos para ordenador si no tienen interfaces compatibles

Interface para cassette
La conexión para cassette o interface de que disponen la mayoría de los ordenadores personales es en realidad un tipo de interface en serie. Debido a que los datos se han de grabar en cintas de cassette normales, utilizando frecuencias de audio, no se pueden conseguir altas velocidades de transferencia de datos. El sistema de circuitos de la interface toma los bytes de datos para grabar de la memoria y convierte cada uno de ellos en un flujo de bits. Al cargar las cintas en la memoria, el sistema de circuitos de la interface decodifica los tonos y los unos y ceros resultantes se ensamblan en bytes de ocho bits para su almacenamiento en la memoria. Las interfaces para cassette de la mayoría de los micros personales son universales, en el sentido de que se puede utilizar cualquier grabadora de cinta normal obteniendo los mejores resultados. El conector empleado no está estandarizado, pero los de uso más común son los conectores DIN o minienchufes.

Conexión en paralelo
Ésta es una interface en paralelo con fines generales para conectar a un micro con dispositivos periféricos. Los ocho bits de cada byte transmitido se envían juntos (en paralelo) a través de ocho cables. Se proporcionan otras señales para sincronizar la transmisión de datos, a fin de que ésta sólo se efectúe cuando el dispositivo que los ha de recibir esté preparado para hacerlo

Interface para unidad de disco
Normalmente las unidades de disco se conectan a los ordenadores utilizando una interface en paralelo. No existe ninguna estandarización y, por regla general, sólo se pueden conectar las unidades de disco fabricadas especialmente para determinado modelo de ordenador



Entrada analógica
Existente por lo general sólo en los ordenadores más caros diseñados con fines educativos, una entrada analógica es útil para conectar el ordenador con dispositivos de laboratorio como sensores de luz o indicadores eléctricos de temperatura. La interface constará sólo de una o más líneas capaces de aceptar y leer un voltaje dentro de una escala determinada. El usuario debe asegurarse de que el ordenador no se conecta a un voltaje que supere dicha escala, lo cual entrañaría peligro

Conexión para ampliación de memoria
Por lo general sustenta la mayor parte de las líneas, si no todas, que vienen directamente desde el microprocesador, es decir, los buses de direcciones, de datos y de control. Es aquí donde se debe enchufar la memoria adicional y, en algunos ordenadores, también los periféricos del fabricante. Puede acoplarse mediante un conector terminal PCB, aunque en algunos casos puede tratarse de un enchufe capaz de acoger un conector terminal, como el de un cartucho para juegos (que es en realidad una forma de ampliación de la ROM)

Conexiones para palanca de mando
No hay ninguna interface estándar para las palancas de mando, aunque muchos fabricantes han adoptado el modelo de Atari. La mayoría de dichas interfaces poseen simplemente cinco líneas activas (una desde cada interruptor de las cuatro modalidades del movimiento de la palanca de mando, y la quinta para el botón de disparo). Sin embargo, las palancas analógicas requieren una interface diferente, capaz de aceptar una amplia escala de voltajes para indicar la posición exacta de la palanca.
La mayoría de los ordenadores disponen de más de una conexión para palanca de mando, si bien en ocasiones varios dispositivos comparten el mismo enchufe

Interface para impresora
Las interfaces para impresoras están relativamente estandarizadas según un sistema desarrollado por la Centronics Corporation, de modo que no es difícil conseguir una impresora con interface Centronics que funcione con la interface para impresora de la mayoría de los ordenadores. Los niveles de señales, así como las funciones de señales, están asimismo estandarizados en 0 y 5 voltios para el 0 y el 1 binarios respectivamente. Los conectores utilizados y la asignación de las señales a las diversas patillas no están estandarizados; por este motivo el usuario tal vez precise de un cable especial para conectarle una impresora al ordenador

Interface para video (RF)
Todos los ordenadores personales están construidos para ser conectados a una unidad de visualización en video, que en la mayoría de los casos es un televisor corriente, en color o en blanco y negro. Si se ha de utilizar el enchufe normal del televisor para la entrada de antena, la señal de video primero ha de ser modulada en RF (radiofrecuencia) para que se asemeje a la señal de transmisión de televisión. Luego es demodulada por el sintonizador del televisor

Interface RGB
Se pueden obtener aún mejores resultados si se mantienen separados los elementos que requiere el monitor de video. La interface RGB proporciona señales de video rojas, verdes y azules separadas, más una señal de sincronización horizontal y otra de sincronización vertical

Interface para video (composite)
Algunos televisores y muchos monitores de video incorporan una entrada de video compuesta que evita la etapa de demodulación de RF, con lo cual se consigue producir imágenes de mayor calidad. En la salida del ordenador están presentes todos los elementos de una señal de video estándar (señales de crominancia, luminancia, sincronización, etc.), pero esta señal "compuesta" no requiere una posterior modulación o procesamiento por parte del sintonizador del televisor



Interface en serie
A diferencia de muchas otras, la interface en serie RS232 está, a nivel teórico, definida con toda precisión según la normativa de la Asociación de Industrias Eléctricas. Ésta especifica el tipo de conector a utilizar (un conector "miniatura D" de 25 patillas), las señales destinadas a cada patilla y los niveles de señales. Lamentablemente, son pocos los fabricantes que se atienen a la normativa y con determinados ordenadores puede resultar difícil hacer funcionar dispositivos en serie. De las muchas patillas que presenta la interface estándar, normalmente sólo se utilizan tres: la 2, para transmitir los datos en serie desde el ordenador al periférico; la 3, para recibir los datos transmitidos desde éste, y la 7, la señal a tierra. Tanto los dispositivos de transmisión como los de recepción se han de ajustar para que la velocidad de transmisión de los datos y el formato de los datos transmitidos sean los mismos

Interface IEEE
Se trata de una interface en paralelo universal basada en el bus de interface Hewlett Packard y adoptada ahora como estándar. La normativa está muy bien definida, tanto física como eléctricamente. A diferencia de otras interfaces para datos (como, por ejemplo, el Centronics en paralelo y el RS232 en serie), que sólo se pueden conectar a un dispositivo a la vez, el bus IEEE se puede acoplar simultáneamente con hasta 15 aparatos (incluyendo el propio ordenador). Los dispositivos que incorporan interfaces IEEE incluyen impresoras, unidades de disco flexible, plotters, generadores de señales, voltímetros y otros equipos de comprobación. Como se presta muy bien para emplearlo con equipos de medición y pruebas, el bus IEEE se utiliza mucho en los laboratorios y establecimientos industriales. En la actualidad son muy pocos los ordenadores personales que ofrecen interfaces IEEE y algunos de ellos utilizan una conexión en el tablero de circuito impreso en lugar del conector IEEE estándar

El término *interface* se aplica con cierta vaguedad para designar la puerta o conector en que se le enchufan al ordenador los dispositivos externos. En sentido estricto, sin embargo, el término se refiere al sistema de circuitos y al software que permiten efectuar la conexión entre dos dispositivos cualesquiera relacionados con el ordenador.

En su interior, el ordenador se comunica enviando datos a través de "buses", grupos de conductores en paralelo, cada uno de los cuales conduce una única señal binaria. En la mayor parte de los microordenadores existen tres buses internos: un bus de datos de 8 bits, un bus de direcciones de 16 bits y un bus de control, cuyas señales, por lo común constan de entre 5 y 12 bits que indican el estado normal de la CPU. Algunas de las señales de control le avisan a la memoria y a los dispositivos periféricos si la CPU desea recibir datos (leer) o depositar datos (escribir). Otras transmiten información desde el exterior a la CPU, informándole, por ejemplo, de que un dispositivo periférico tiene que dar entrada a determinados datos y requiere atención.

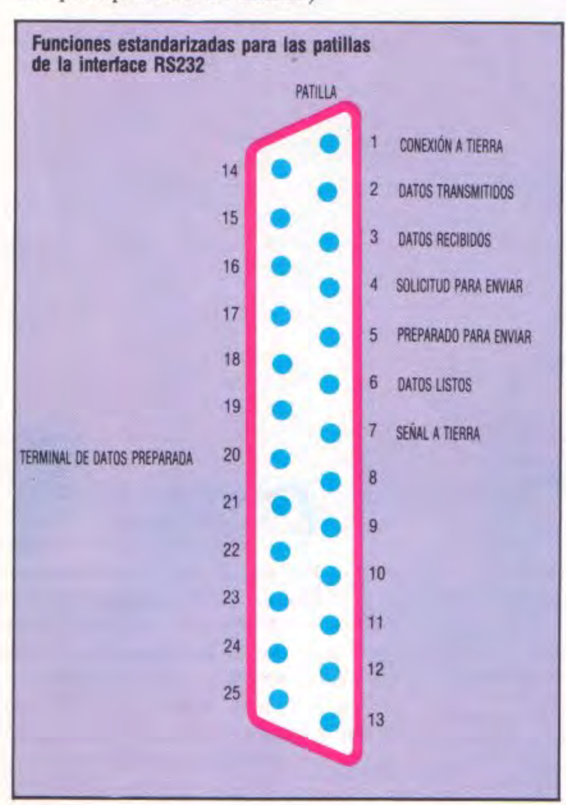
Internamente, el ordenador por lo general manipula la información compuesta ya sea de 8 bits o de 16 bits a la vez. Por tanto, si la CPU desea recuperar los datos de la dirección de memoria 65535 (o FFFF, expresada en hexadecimal), establecerá todos los 16 cables del bus de direcciones en uno para identificar esa localización. Si el contenido de esta dirección de memoria resultara ser 182 (B6, en hexadecimal), este dato se colocaría en el bus de datos como los ocho dígitos binarios 10110110.

Cuando se transfieren datos de esta manera, de a 8 o 18 bits a la vez, se dice que la transferencia es "en

paralelo". Muchos dispositivos periféricos también están diseñados para transmitir o recibir datos en paralelo. Los interfaces que se facilitan para los periféricos de este tipo se denominan *interfaces en paralelo* y la mayoría de los ordenadores proporcionan al menos un conector específicamente para dispositivos "en paralelo".

No todos los dispositivos periféricos pueden recibir o transmitir datos en paralelo. Algunos se valen de un solo cable para comunicarle al ordenador un bit a la vez. Internamente siguen utilizando los datos en forma de bytes de 8 o 16 bits, pero los bits de cada byte se transmiten o se reciben de a uno cada vez, empezando por el bit "menos significativo" del byte y terminando con el bit "más significativo". Cada byte se divide en un flujo de bits, enviados uno después del otro, y vueltos a ensamblar en un byte al otro extremo, utilizando unos circuitos especiales convertidores de paralelo a en serie y viceversa.

Se puede hacer que tanto las interfaces en serie como las interfaces en paralelo transmitan información ya sea desde el ordenador o hacia el mismo. Los ordenadores suelen incluir otras interfaces, que o bien siempre envían la información hacia afuera (por ejemplo, el circuito de salida del televisor) o siempre aceptan la información entrante (por ejemplo, las conexiones para palancas de mando).



Interfaces en serie

Existe una interface en serie estandarizada, la RS232, para la cual están definidos con todo detalle los niveles de señales y la asignación de patillas. Hasta el tipo de conector está especificado. Lamentablemente, son pocos los fabricantes que se adhieren por completo a la normativa y hacer funcionar las conexiones en serie puede resultar difícil. La RS232 es una interface en serie "bidireccional", con una patilla (terminal) para transmitir datos desde el ordenador y otra destinada a recibirlos.

Los datos se envían de a un bit cada vez por vía del terminal de "datos transmitidos" y se reciben a través del terminal de "datos recibidos". El flujo de bits puede asumir diversos formatos estandarizados, pero no importa cuál de ellos se utilice siempre y cuando ambos dispositivos, el de recepción y el de transmisión, empleen el mismo.

Dado que cada byte de información se envía hacia afuera como un flujo de bits en serie, el software que controla a la interface ha de tener algún modo de indicar cuándo comienza el primer bit del dato y cuándo ha terminado el último bit. El procedimiento utilizado más comúnmente tiene un único "bit de inicio" (0, en lógica de Boole), seguido de ocho bits de datos, seguidos de un único "bit de parada" (un 1 lógico).

Es necesario especificar de antemano la velocidad a la cual se transmiten los datos, ya que, de lo contrario, se podría malinterpretar el patrón de impulsos que representa en forma de ceros y unos a los ocho bits de datos. Esta velocidad de transmisión de datos se denomina *velocidad baudio*, en honor del francés Baudot, que la inventara en el siglo XIX. Las velocidades baudio oscilan entre 75 y 9 600 baudios. Estas cifras corresponden a los 75 y 9 600 bits por segundo que se transmiten, y como normalmente hay 10 bits (incluyendo los bits de inicio y de parada) para cada carácter, la velocidad de transmisión de caracteres equivale a la décima parte de la velocidad baudio.

Interfaces en paralelo

La interface en paralelo transmite o recibe la información de a un byte cada vez, pero además de las ocho "líneas de datos" necesita también proporcionar otras señales para que el ordenador y el periférico sepan cuándo es posible transmitir datos y cuándo no. El tipo más común de interface en paralelo es la "Centronics" (llamada así en razón de la fábrica norteamericana de impresoras Centronics Corporation); pero este supuesto modelo estándar no está muy afianzado. El tipo de conector que se usa y la asignación de señales a las diversas patillas presentan muchas diferencias de un fabricante a otro. La mayoría de interfaces Centronics proporcionan al menos las señales siguientes:

De DATO 0 a DATO 7	Ocho cables para transportar los ocho bits del byte que se está transmitiendo
ADK	Una señal de entrada para el ordenador que indica que el dispositivo receptor está preparado para aceptar datos
GND	El cable "a tierra" que proporciona una referencia común de 0 voltios tanto al ordenador como al dispositivo periférico
BUSY	Una señal desde el dispositivo periférico hasta el ordenador que indica que el periférico no puede aceptar información
STROBE	Una señal de salida desde el ordenador que le indica al periférico que la información está preparada y debe leerla

Muchos otros dispositivos, aparte de las impresoras, adoptan la interface en paralelo casi estandarizada de Centronics, y es probable que para conectarlas al ordenador sólo haya que comprar un cable especial para conexión o acondicionar uno el mismo usuario.

Entregado a la suerte

Los ordenadores son totalmente racionales, y, en consecuencia, incapaces de elegir un número al azar, que es una tarea irracional

"Elija un número al azar." Ésta es una de las cosas más fáciles para cualquiera; pero, paradójicamente, esta sencilla tarea es imposible para un ordenador.

Un número al azar, o aleatorio, es un número imposible de predecir en cualquier circunstancia. Un ordenador no hace más que seguir instrucciones y necesita una razón para todas las acciones que asume, de modo que cualquier número que cree será la consecuencia de una serie de instrucciones. Independientemente del grado de complejidad de estas instrucciones, el número será aún teóricamente predecible y por ello el número no es auténticamente aleatorio.

Sin embargo, se pueden obtener auténticos números aleatorios a partir de procesos físicos. Por ejemplo, en Gran Bretaña, el ERNIE (*Electronics Random Number Indicator Equipment:* equipo electrónico indicador de números al azar) utiliza el movimiento al azar de electrones libres para elegir los números ganadores de los bonos emitidos por las cajas de ahorros y que participan en una especie de lotería nacional.

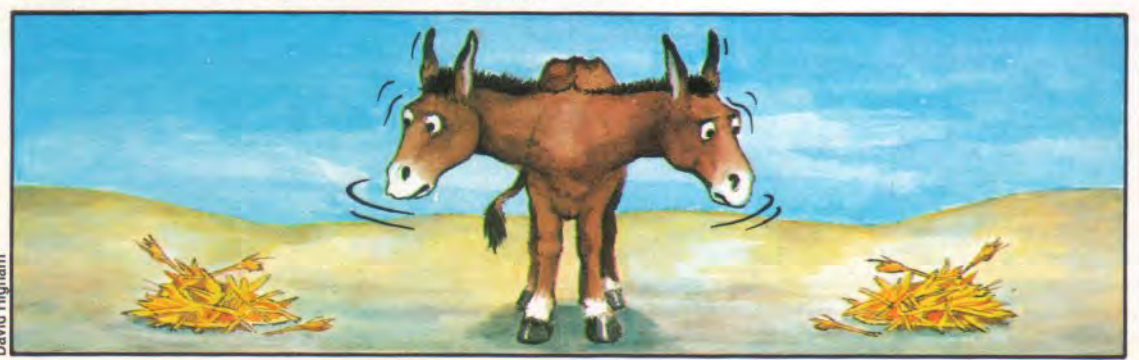
Fue Von Neumann quien concibió la idea de los números seudoaleatorios, que se podían crear matemáticamente. Su método consistía en tomar un número de cuatro dígitos, por ejemplo, el 4321, y elevarlo al cuadrado. Del resultado obtenido de ocho dígitos, 18671041, él tomaba los cuatro dígitos centrales, 6710, como el número al azar. Éste se convierte luego en la "semilla" a partir de la cual se creará el número siguiente. De manera que 6710 al cuadrado da 45024100, cuyos dígitos centrales son 0241, que se convierte en el número aleatorio, y así sucesivamente. Este proceso puede seguir indefinidamente, pero dado que el total de los diferentes números posibles con cuatro dígitos es limitado (9999), tarde o temprano la secuencia habrá de empezar a repetirse.

Los ordenadores modernos utilizan procedimientos más complicados que proporcionan números seudoaleatorios al instante. Por ejemplo, el micro BBC puede generar un número al azar aproximadamente cada 1,6 milisegundos, y si generara estos números de forma continuada e ininterrumpida, transcurrirían 150 días hasta que la secuencia se repitiera.

Los números aleatorios se generaron por primera vez electrónicamente para que los emplearan los ingenieros de telecomunicaciones cuando intentaban simular las fluctuaciones de la demanda en las centrales telefónicas. En la actualidad se emplean para muchos fines, desde la simulación de procesos que fluctúan al azar en los juegos por ordenadores, hasta la evaluación de difíciles funciones matemáticas. Para la mayoría de las aplicaciones, los algoritmos para números aleatorios se pueden considerar genuinamente al azar.

Kevin Jones

El método Monte Carlo
Este método lo inventó John von Neumann y se vale de los números al azar para calcular las respuestas a problemas matemáticos. En matemáticas con frecuencia se necesita calcular la superficie delimitada por una curva y esto, en cierto sentido, es análogo a hallar la superficie de una isla cuando sólo se dispone de un mapa de las líneas costeras. Se ilustra el mapa de Gran Bretaña en un recuadro de dimensiones conocidas, que será bombardeado por puntos al azar. Dado que las posiciones de los puntos se generan aleatoriamente, caerán sobre toda la superficie enmarcada en el recuadro, y el número que caiga dentro de la isla será proporcional a la superficie de Gran Bretaña. Utilizando 40 números aleatorios hallamos que 24 caen en el mar y 16 en tierra. Por tanto, la superficie de la isla es: $16/40 \times (1050 \times 550) = 16/40 \times 577\,500 = 231\,000$ kilómetros cuadrados.
Si se utilizara un mayor número de puntos al azar el resultado se aproximaría más a la cifra de 229 523 km^2, superficie de Gran Bretaña



David Higham

El asno racional
Imaginemos que colocamos a un borrico totalmente racional a medio camino entre dos pilas de heno idénticas. El borrico no puede escoger la pila más grande porque ambas son iguales, y no puede escoger la que esté más cerca porque él está exactamente en el medio de las dos. De modo que tal vez el animal se dirigiera hacia la que hubiera visto primero. Pero ¿por qué habría de mirar a una en lugar de a la otra? Puesto que no existe razón alguna que le haga elegir una pila y no la otra, el asno simplemente se quedaría allí, en el medio, y se moriría de hambre. Del mismo modo, los ordenadores no pueden generar números auténticamente aleatorios, porque todo lo que hacen lo realizan estrictamente de acuerdo a la razón

Sinclair ZX81

En 1981 este pequeño ordenador se consideró revolucionario, por su diseño y su precio. Desde entonces su precio ha bajado aún más, no así su calidad

El Sinclair ZX 81 es el ordenador más barato con visualización en pantalla y es una versión perfeccionada del ZX80. El número de chips queda reducido por la utilización de un chip ULA (*Uncommitted Logic Array*: disposición lógica no comprometida) para manipular las funciones que previamente requerían muchos componentes (véase recuadro). Es como el Ford modelo T de los ordenadores personales: negro, sin teclas que sobresalgan, ni interruptores. La fuente de alimentación eléctrica externa se enchufa a un lado de la carcasa cuneiforme. Las otras conexiones son para cassette, cable de televisión y periféricos. La impresora Sinclair ZX es, asimismo, el medio más económico de obtener "copias impresas" o listados, si bien el papel plateado electrosensible que emplea es relativamente caro. La pantalla del ZX81, a diferencia de otras visualizaciones monocromáticas, imprime caracteres negros sobre un fondo blanco y sólo en la parte superior de la pantalla. El BASIC de Sinclair difiere de la versión más estandarizada, la Microsoft, pero es muy adecuado para el principiante.

El teclado del ZX81 es sorprendente: consiste en la imagen impresa de un teclado, debajo del cual hay un relleno de teclas sensibles al ser pulsado aquél. Por ello no es adecuado para la escritura al tacto, si bien en cierta medida este inconveniente queda superado al introducir los programas. Cuando el ordenador se enciende por primera vez, el cursor se visualiza como una K. Esto significa que una "palabra-tecla" como PRINT o LET se imprimirá al completo en la pantalla al pulsar los botones P o L. Estas órdenes figuran escritas arriba de cada tecla. De forma similar, el cursor se puede colocar en "modalidad de función" (visualizándose una F). Estando en esta modalidad, al pulsar una tecla se imprimirá la palabra de la función impresa arriba o debajo de esa tecla. Para dar entrada a un texto normal está la L o modalidad "de letras".

A pesar de ser pequeño y barato, el ZX81 aún se puede considerar, con toda justicia, como un ordenador, y lleva en el mercado el tiempo suficiente como para que se haya desarrollado alrededor de él una amplia gama de software. Los entusiastas de la informática han tomado el diminuto Kbyte de su memoria estándar como un desafío y se las han ingeniado para comprimir programas de aventuras y de ajedrez, por ejemplo. Se pueden comprar 16 Kbytes de memoria extra en forma de una pequeña caja de plástico que se enchufa en la parte posterior del ordenador. Sin embargo, este dispositivo le ha causado a algunos programadores considerables molestias, porque el conector no es muy seguro y el más ligero movimiento puede significar la pérdida de toda la información. Algunos usuarios del ZX81 descubrieron que la cinta adhesiva de doble faz resultaba muy útil para solucionar este inconveniente, pero Sinclair Research ha sustituido el antiguo paquete de RAM por uno más fiable.

El ZX81 también ha dado lugar a la proliferación de pequeñas empresas que ofrecen toda clase de extras. Algunas de ellas producen paquetes de RAM más grandes y más fiables, conectores para las impresoras de marca distinta a Sinclair, y visualizaciones en color. Existen incluso carcasas de recambio que incorporan mejores teclados y espacio para alojar en su interior la mayoría de los extras y accesorios. La conversión es sólo cuestión de quitar el PCB de la carcasa original y colocar el nuevo en su lugar.

Con todo, el Sinclair ZX81 es el medio más barato que existe para introducirse en el mundo de la informática, y es asimismo la máquina ideal para quien no desee invertir mucho dinero en la adquisición de un ordenador personal. Además, a los niños les encanta su teclado. En los dos últimos años la popularidad de esta máquina ha aumentado de forma considerable.

El teclado del ZX81
Con el fin de producir un microordenador con teclado integral por un precio económico, Sinclair adoptó una "membrana múltiple sellada", que desde entonces ha sido muy criticada por su insensibilidad y su falta de "tacto". La necesidad de disponer el teclado en un espacio pequeño determinó que hubieran de emplearse técnicas de optimización, que cada tecla individual cumpla diversos cometidos. La tecla L, por ejemplo, devuelve ese carácter, la asignación LET, la función USR o el signo =, según la modalidad que se utilice en un momento dado

ULA
Sinclair lo llama el chip de "lógica Sinclair para ordenadores". Es un chip especial que contiene el equivalente de diversos chips "lógicos", más pequeños. En el lenguaje informático éstos poseen un nombre, chips "encolados", porque "pegan" entre sí a los componentes importantes de un ordenador. La inclusión de este ULA marca la diferencia entre el ZX80 y el ZX81. En el ZX80 hay alrededor de 12 pequeños chips "encolados", lo que significa que su construcción es más costosa y tiene más probabilidades de fallo. Los ULA, sin embargo, son difíciles de realizar y, al igual que las unidades ROM, el diseño debe ser definitivo y completo antes de que se puedan fabricar

Modulador de RF
Para aparatos de televisión corrientes. Proporciona salida de video

Cantidad contra calidad
Cuando sir Clive Sinclair decidió lanzar al mercado de los ordenadores personales un micro de bajo costo, se vio obligado a transigir respecto a ciertas características en el diseño del aparato. Quizá sea en el teclado donde más se advierta este hecho, pues durante mucho tiempo se lo ha venido considerando como el punto más débil tanto del ZX80 como del 81. Como resultado de ello, muchos usuarios llegaron a creer que la máquina era ideal para ejecutar software "comprado", más que para dar entrada a sus propios programas más largos, y rápidamente salieron a la venta muchos programas, en particular para el mercado de juegos. Pisándole los talones a las inevitables versiones de Breakout, Space Invaders y gusanitos que se comen entre sí, vinieron los medios auxiliares para la eliminación de errores del código de lenguaje máquina y, sorprendentemente, una variedad de paquetes de gestión, en especial hojas electrónicas, si bien éstos requieren del paquete de RAM adicional. El BASIC no es el único lenguaje disponible para el ZX81. Sinclair ofrece, además, FORTH (véase p. 150) y un ensamblador para el microprocesador Z80

Conectores
Tres conectores en miniatura para alimentación eléctrica, entrada y salida de cassette. Existen algunos programas que producen música desde el conector para salida de cassette





Chris Stevens

CPU
El ZX81 utiliza un microprocesador Z80 estándar

Conector para teclado
Las dos hileras de patillas metálicas en la parte frontal derecha del tablero son para conectar el teclado al tablero de circuitos. Son las líneas en hilera y en columna del teclado

RAM
El tablero está diseñado para permitir la utilización de diversas clases de chips, dependiendo del tipo del cual disponga Sinclair en el momento de la fabricación. De manera que la RAM puede ser un chip grande o, como muestra la ilustración, dos chips más pequeños

ROM
Contiene todo el software incorporado: el generador de caracteres para la visualización en pantalla, el lenguaje BASIC y el sistema operativo para cargar y salvar la cassette. Si la ROM de Sinclair hubiera de ser leída por otro ordenador basado en el Z80, sus contenidos no tendrían el aspecto de ser instrucciones para esta máquina. Ello se explica porque se han mezclado algunas de las líneas de datos para la ROM, debido quizás al deseo de dificultar las posibles copias o, sencillamente, de simplificar el trazado del tablero de circuitos

Conector terminal
En el conector terminal, situado en la parte posterior del tablero del circuito, se enchufan la memoria y todos los otros periféricos

ZX81

DIMENSIONES
175 × 168 × 43 mm

PESO
300 g

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
ROM de 8 K que contiene el intérprete de BASIC y el sistema operativo. RAM de 1 K de la cual se toman 123 bytes para las variables del sistema y el resto lo comparten su programa en BASIC y la visualización en pantalla. La RAM es ampliable hasta 16 K

VISUALIZACION EN VIDEO
Blanco y negro, líneas de texto de 32 × 24, gráficos de 64 × 44. Las dos líneas inferiores no se usan en una visualización normal: muestran lo que se está digitando en el ordenador

INTERFACES
Los conectores estándar son I/O cassette, entrada de alimentación eléctrica y salida de televisión. El conector terminal posterior acepta otros periféricos y memoria extra

LENGUAJE SUMINISTRADO
BASIC de Sinclair

OTROS LENGUAJES DISPONIBLES
FORTH, ensamblador para el código de lenguaje máquina Z80

VIENE CON
Adaptador para fuente de alimentación eléctrica, cable para televisor, cable para cassette, documentación

TECLADO
40 espacios sensibles al tacto, cada uno de ellos capaz de diversos cometidos: orden en BASIC, un carácter alfanumérico o un símbolo gráfico

DOCUMENTACION
El manual, sencillo, es una introducción muy clara y concisa a la forma de utilizar este ordenador. Parte de la premisa de que el lector no posee ninguna clase de conocimientos previos de informática, y ofrece ejemplos, ejercicios y resúmenes al final de cada capítulo.A partir del capítulo 23 se describe el ordenador con toda clase de detalles técnicos y se proporcionan indicaciones acerca de la forma de utilizar el código de lenguaje máquina y una lista completa de todas las variables que utiliza el sistema operativo

Estructuras de control

Todas las versiones de BASIC incorporan estructuras que gobiernan el flujo de un programa. No obstante, algunas máquinas ofrecen una amplia gama de alternativas, con sutiles diferencias

En los diez primeros capítulos del curso de programación BASIC hemos cubierto casi todos los aspectos más importantes del lenguaje BASIC. En esta ocasión le ofreceremos una recopilación de los temas que hemos desarrollado hasta ahora, haremos algunas interesantes digresiones y proporcionaremos al lector de la obra una idea general de los temas que nos proponemos desarrollar en adelante.

Dediquémonos primero al resumen: un lenguaje de alto nivel como el BASIC le proporciona al usuario una serie de instrucciones que se traducen internamente en una forma comprensible para el ordenador. Todo programa para ordenador se puede escribir utilizando tan sólo dos sencillos patrones denominados *construcciones*. Se trata de las "construcciones de secuencia" y de las "estructuras de control", de las cuales sólo dos son esenciales en BASIC: IF...THEN...ELSE y WHILE...DO. La mayor parte de los otros lenguajes para ordenador proporcionan una cantidad de instrucciones considerablemente superior.

La construcción de secuencia permite que la tarea se divida en una serie de subtareas que, al ser ejecutadas en secuencia, efectúan la tarea principal. Las dimensiones de las subtareas dependen del lenguaje; en BASIC las subtareas se representan mediante las sentencias escritas en cada línea, y la secuencia se representa, a su vez, por medio de los números de línea. Por lo tanto, si la tarea consistiera en multiplicar por 10 el valor asignado a una variable, la secuencia que utilizaríamos podría ser la siguiente:

```
110 INPUT N
120 LET N = N * 10
130 PRINT N
```

Además de construcciones de secuencia necesitamos también estructuras de control. Se trata de construcciones que alteran el orden de ejecución de las sentencias de un programa.

La estructura de control más sencilla que proporciona el BASIC es GOTO. Se trata de un salto (o bifurcación) incondicional que redirige la ejecución del programa hasta un número de línea determinado sin que se haya de satisfacer ninguna prueba o condición. GOSUB es también una bifurcación incondicional, pero el programa siempre retornará (RETURN) hasta el punto inmediatamente posterior a la GOSUB y su utilización en la programación estructurada es perfectamente aceptable.

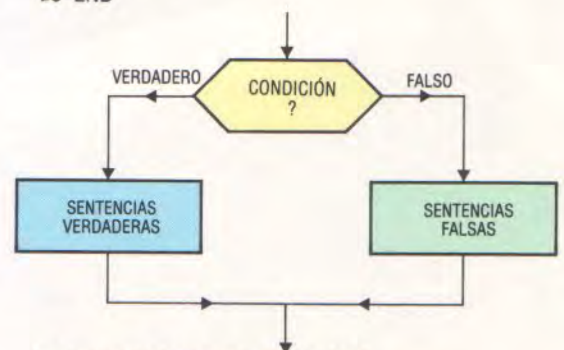
El BASIC dispone de la estructura de control IF... THEN...ELSE. Ésta asume la forma de la sentencia IF...THEN y posee la siguiente sintaxis (téngase presente que, en el lenguaje informático, la palabra "sintaxis" equivale a "forma"):

Si (IF) (la condición especificada) es verdadera *entonces* (THEN) ejecute la sentencia especificada (*si no*) (ELSE), debe ejecutar la sentencia siguiente.

Observe que en el BASIC estándar, la palabra ELSE de IF...THEN...ELSE está implícita. En algunas versiones de BASIC y en algunos otros lenguajes, como el PASCAL, por ejemplo, ELSE forma parte de la sentencia.

IF...THEN...ELSE (IF...THEN en BASIC) realiza una de dos subtareas según si una determinada condición es verdadera o no. Consideremos el siguiente programa, cuyo propósito es hallar la raíz cuadrada de números que se digitan por el teclado. La entrada del número −9999, por ejemplo, nos indicará que queremos terminar el programa:

```
10 PRINT "DE ENTRADA A UN NUMERO"
20 INPUT N
30 IF N = −9999 THEN GOTO 70
40 LET S = SQR(N)
50 PRINT "LA RAIZ CUADRADA DE";N;"ES";S
60 GOTO 10
70 END
```



La estructura de control IF...THEN...ELSE
Si la condición es verdadera, se ejecutarán las sentencias verdaderas. Si la condición es falsa, se ejecutarán las sentencias falsas

Aquí lo que la línea 30 dice realmente es "*Si* (IF) es verdad que N = −9999, *entonces* (THEN) vaya hasta el final del programa, *si no* (ELSE) (si no es verdad que N = −9999) ejecutar la siguiente línea del programa para hallar la raíz cuadrada".

El BASIC no dispone directamente de la otra estructura de control esencial (WHILE...DO), pero se puede conseguir fácilmente. WHILE...DO es una forma de "hacer un bucle" y significa "repita una sentencia o serie de sentencias *mientras* (WHILE) una condición sea verdadera, *haga* (DO) algo".

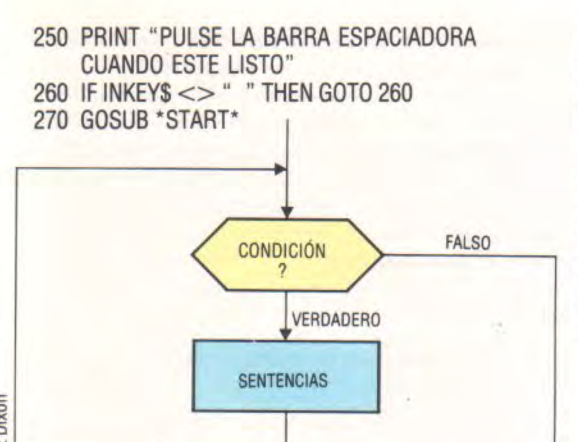
WHILE...DO siempre comprueba la condición antes de que se ejecuten las sentencias, de manera que si la comprobación fracasa la primera vez que se realiza, las sentencias (denominadas el cuerpo del bucle) no se ejecutan. Como ejemplo, consideremos un programa para juegos que le indica al jugador que 'PULSE LA BARRA ESPACIADORA CUANDO ESTE LISTO'. Esta parte se podría escribir asimismo (usando un seudolenguaje o castellano simplificado) de la manera siguiente:

Mientras (WHILE) no se pulse la barra espaciadora
explore (DO) el teclado
comienza juego

En BASIC esto se escribiría:



```
250 PRINT "PULSE LA BARRA ESPACIADORA
    CUANDO ESTE LISTO"
260 IF INKEY$ <> " " THEN GOTO 260
270 GOSUB *START*
```



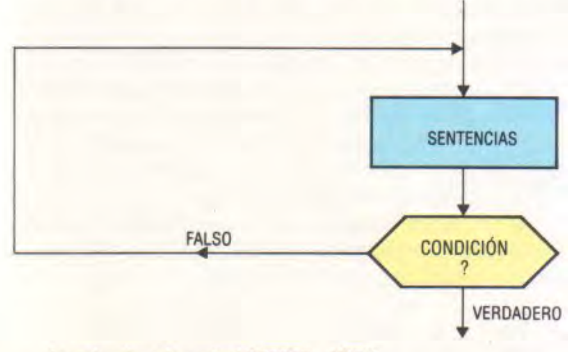
La estructura de control DO...WHILE
El bucle se repite en la medida en que la condición es verdadera. Es posible que las sentencias no lleguen a ejecutarse nunca (si la condición inicial es falsa)

La línea 260 dice que *si* (IF) INKEY$ no es igual a (<>) un espacio (" "), *entonces* (THEN), retroceda y verifique nuevamente el teclado. Una forma algo más elegante de escribir esto sería la siguiente:

```
250 PRINT "PULSE LA BARRA ESPACIADORA
    CUANDO ESTE LISTO"
260 FOR X = 0 TO 1 STEP 0
270 IF INKEY$ = " " THEN LET X = 2
280 NEXT X
290 GOSUB * STAR *
```

En este fragmento del programa el bucle (para explorar el teclado) sólo se ejecuta si no se ha pulsado la barra espaciadora. Si se ha pulsado (es decir, INKEY$ = " "), entonces el programa abandona el bucle FOR...NEXT y va a la línea 290, que llama a la subrutina START. (Obsérvese que estamos utilizando "etiquetas" o nombres para las subrutinas. Muchas versiones de BASIC no pueden llamar a las subrutinas por un nombre; en tal caso, el usuario tendrá que emplear, en lugar de las etiquetas, números de línea.)

Hasta ahora nunca nos habíamos encontrado con STEP y quizá ésta sea una forma algo inusual de aplicarla. Cuando se utiliza un bucle FOR...NEXT, STEP permite que el "índice" se incremente en unidades distintas a uno. FOR I = 1 TO 10 STEP 2 hará que I tenga el valor 1 la primera vez que se efectúe el bucle, y posteriormente 3, 5, 7 y 9. El incremento siguiente (a 11) superaría el límite de 10, de modo que el bucle se da por concluido. Incluso es posible hacer que el índice cuente hacia atrás. FOR I = 10 TO 1 STEP–1 hará que I cuente hacia atrás desde 10 hasta 1. Utilizar STEP 0 es realmente un truco inteligente que asegura que el bucle nunca terminará a menos que X "se incremente artificialmente", como en el caso de nuestra sentencia IF...THEN.



La estructura de control REPEAT...UNTIL
El bucle se repite hasta que la condición resulta verdadera. Las sentencias siempre se ejecutarán al menos una vez

Otra estructura de control muy útil que tampoco está disponible en BASIC directamente pero que se puede imitar con facilidad, es REPEAT...UNTIL. En este caso la prueba de condición viene después del cuerpo principal del bucle, de manera que la sentencia o las sentencias del cuerpo principal siempre se repetirán al menos una vez. Observemos este "generador de números al azar":

```
10 PRINT "PULSE LA BARRA ESPACIADORA"
20 FOR X = 0 TO 1 STEP 0
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = " " THEN LET X = 2
60 NEXT X
70 PRINT "EL VALOR DE R ES   ";R
```

Aquí, el cuerpo principal (que incrementa el valor de R) siempre se ejecuta al menos una vez, dado que la prueba para la bifurcación del bucle (IF INKEY$ = " ") llega sólo después de la sentencia de incremento (LET R = R + 1).

Otra estructura de control muy útil pero, sin embargo, no esencial, es la que generalmente se denomina CASE. En BASIC, la estructura CASE se aplica empleando ON...GOTO o bien ON...GOSUB. Funciona de esta manera. ON...GOTO es una sentencia de bifurcación múltiple que incorpora varias pruebas condicionadas IF...THEN en una sola sentencia. Consideremos, por ejemplo, un fragmento de programa que convierta los números del 1 al 7 en los nombres de los siete días de la semana:

```
1050 IF D = 1 THEN GOTO 2020
1060 IF D = 2 THEN GOTO 2040
1070 IF D = 3 THEN GOTO 2060
1080 IF D = 4 THEN GOTO 2080
1090 IF D = 5 THEN GOTO 3000
2000 IF D = 6 THEN GOTO 3020
2010 IF D = 7 THEN GOTO 3040
2020 PRINT "LUNES"
2030 GOTO*END*
2040 PRINT "MARTES"
2050 GOTO*END*
2060 PRINT "MIERCOLES"
2070 GOTO*END*
2080 PRINT "JUEVES"
2090 GOTO*END*
3000 PRINT "VIERNES"
3010 GOTO*END*
3020 PRINT "SABADO"
3030 GOTO*END*
3040 PRINT "DOMINGO"
3050 GOTO*END*
```

En BASIC, una forma más concisa de conseguir los mismos resultados sería la de utilizar ON...GOTO de la siguiente manera:

```
1050 ON D GOTO 2020, 2040, 2060, 2080,
         3000, 3020, 3040
```

ON...GOSUB funciona igual excepto que el valor de la variable determina hacia qué subrutina se bifurca. A continuación le ofrecemos una ligera modificación del programa de los dados (véase p. 174) utilizando ON...GOSUB para seleccionar los gráficos adecuados para los dados seleccionados por la función RND:

DECIMAL	BINARIO		CARACTER
32	0 0 1 0 0 0 0 0	=	espacio
33	0 0 1 0 0 0 0 1	=	!
34	0 0 1 0 0 0 1 0	=	"
35	0 0 1 0 0 0 1 1	=	#
36	0 0 1 0 0 1 0 0	=	$
37	0 0 1 0 0 1 0 1	=	%
38	0 0 1 0 0 1 1 0	=	&
39	0 0 1 0 0 1 1 1	=	'
40	0 0 1 0 1 0 0 0	=	(
41	0 0 1 0 1 0 0 1	=	)
42	0 0 1 0 1 0 1 0	=	*
43	0 0 1 0 1 0 1 1	=	+
44	0 0 1 0 1 1 0 0	=	,
45	0 0 1 0 1 1 0 1	=	-
46	0 0 1 0 1 1 1 0	=	.
47	0 0 1 0 1 1 1 1	=	/
48	0 0 1 1 0 0 0 0	=	0
49	0 0 1 1 0 0 0 1	=	1
50	0 0 1 1 0 0 1 0	=	2
51	0 0 1 1 0 0 1 1	=	3
52	0 0 1 1 0 1 0 0	=	4
53	0 0 1 1 0 1 0 1	=	5
54	0 0 1 1 0 1 1 0	=	6
55	0 0 1 1 0 1 1 1	=	7
56	0 0 1 1 1 0 0 0	=	8
57	0 0 1 1 1 0 0 1	=	9
58	0 0 1 1 1 0 1 0	=	:
59	0 0 1 1 1 0 1 1	=	;
60	0 0 1 1 1 1 0 0	=	<
61	0 0 1 1 1 1 0 1	=	=
62	0 0 1 1 1 1 1 0	=	>
63	0 0 1 1 1 1 1 1	=	?
64	0 1 0 0 0 0 0 0	=	@
65	0 1 0 0 0 0 0 1	=	A
66	0 1 0 0 0 0 1 0	=	B
67	0 1 0 0 0 0 1 1	=	C
68	0 1 0 0 0 1 0 0	=	D
69	0 1 0 0 0 1 0 1	=	E
70	0 1 0 0 0 1 1 0	=	F
71	0 1 0 0 0 1 1 1	=	G
72	0 1 0 0 1 0 0 0	=	H
73	0 1 0 0 1 0 0 1	=	I
74	0 1 0 0 1 0 1 0	=	J
75	0 1 0 0 1 0 1 1	=	K
76	0 1 0 0 1 1 0 0	=	L
77	0 1 0 0 1 1 0 1	=	M
78	0 1 0 0 1 1 1 0	=	N
79	0 1 0 0 1 1 1 1	=	O
80	0 1 0 1 0 0 0 0	=	P
81	0 1 0 1 0 0 0 1	=	Q
82	0 1 0 1 0 0 1 0	=	R
83	0 1 0 1 0 0 1 1	=	S
84	0 1 0 1 0 1 0 0	=	T
85	0 1 0 1 0 1 0 1	=	U
86	0 1 0 1 0 1 1 0	=	V
87	0 1 0 1 0 1 1 1	=	W
88	0 1 0 1 1 0 0 0	=	X
89	0 1 0 1 1 0 0 1	=	Y
90	0 1 0 1 1 0 1 0	=	Z
91	0 1 0 1 1 0 1 1	=	[
92	0 1 0 1 1 1 0 0	=	\
93	0 1 0 1 1 1 0 1	=	]
94	0 1 0 1 1 1 1 0	=	^
95	0 1 0 1 1 1 1 1	=	
96	0 1 1 0 0 0 0 0	=	`
97	0 1 1 0 0 0 0 1	=	a
98	0 1 1 0 0 0 1 0	=	b
99	0 1 1 0 0 0 1 1	=	c
100	0 1 1 0 0 1 0 0	=	d
101	0 1 1 0 0 1 0 1	=	e
102	0 1 1 0 0 1 1 0	=	f
103	0 1 1 0 0 1 1 1	=	g
104	0 1 1 0 1 0 0 0	=	h
105	0 1 1 0 1 0 0 1	=	i
106	0 1 1 0 1 0 1 0	=	j
107	0 1 1 0 1 0 1 1	=	k
108	0 1 1 0 1 1 0 0	=	l
109	0 1 1 0 1 1 0 1	=	m
110	0 1 1 0 1 1 1 0	=	n
111	0 1 1 0 1 1 1 1	=	o
112	0 1 1 1 0 0 0 0	=	p
113	0 1 1 1 0 0 0 1	=	q
114	0 1 1 1 0 0 1 0	=	r
115	0 1 1 1 0 0 1 1	=	s
116	0 1 1 1 0 1 0 0	=	t
117	0 1 1 1 0 1 0 1	=	u
118	0 1 1 1 0 1 1 0	=	v
119	0 1 1 1 0 1 1 1	=	w
120	0 1 1 1 1 0 0 0	=	x
121	0 1 1 1 1 0 0 1	=	y
122	0 1 1 1 1 0 1 0	=	z
123	0 1 1 1 1 0 1 1	=	{
124	0 1 1 1 1 1 0 0	=	|
125	0 1 1 1 1 1 0 1	=	}
126	0 1 1 1 1 1 1 0	=	~

ASCII
He aquí una lista completa de los valores de ASCII entre 32 y 126, sus equivalentes en binario y los caracteres que representan. El significado atribuido a los valores fuera de esta escala varía considerablemente de una máquina a otra

```
390 REM SUBRUTINA DE SELECCION
400 REM UTILIZANDO ON...GOSUB
410 ON D GOSUB 530, 600, 670, 740, 810, 880
470 RETURN
```

Es posible que su versión de BASIC contenga muchas sentencias y funciones que no hayamos explicado; la mayoría de ellas corresponden a ampliaciones del BASIC "básico", concebidas para sacar más partido a determinadas configuraciones de su máquina. Muchas de ellas estarán relacionadas con las configuraciones para gráficos incorporadas en el hardware: instrucciones como PAINT, PAPER, INK, BEEP y CIRCLE. Éstas tienden a ser "específicas de la máquina" y por ello no las hemos incluido en nuestro curso, si bien en futuros capítulos le proporcionaremos más detalles.

No obstante, antes de dar por terminada la parte de nuestro curso de BASIC, ataremos algunos cabos sueltos: analizaremos el juego de caracteres ASCII, estudiaremos un par de funciones que ayudan a manejar los caracteres y veremos una manera de definir nuevas funciones (o funciones no incluidas en su versión de BASIC).

Con el correr de los años se han ido desarrollando diversos métodos para representar las letras del alfabeto y otros caracteres, como los números y los signos de puntuación, en forma digital. Uno de los primeros fue el código Morse, que utiliza puntos y rayas para representar los caracteres. Desde el punto de vista del ordenador, el código Morse tiene el inconveniente de que emplea distintas cantidades de bits para diferentes letras: entre uno y seis puntos y rayas para cada carácter. Otros intentos tendentes a elaborar un código de caracteres más regular y sistemático (por ejemplo, el código Baudot, que utiliza cinco bits para representar hasta 32 caracteres) han fracasado en algún punto y, en la actualidad, el sistema que se ha adoptado con carácter casi universal es el código ASCII (*American Standard Code for Information Interchange*: código norteamericano estándar para intercambio de información).

El código ASCII emplea sólo un byte para representar los 94 caracteres imprimibles, el "espacio" y un número de "caracteres" de control. Ocho bits podrían dar 256 (2^8) combinaciones únicas, pero esto es mucho más de lo que se necesita para representar los caracteres del teclado de una máquina de escribir o un ordenador corrientes, de modo que sólo se utilizan siete, que permiten 128 combinaciones exclusivas. (El octavo bit normalmente no se usa, pero en algunas ocasiones se emplea para especificar un juego alternativo en un idioma diferente o un juego de caracteres para gráficos.) En la tabla que incluimos en la p. 213 se proporcionan los códigos ASCII binario y decimal para la gama estándar de caracteres.

Como puede observar a partir de la tabla, el código ASCII para la letra A es 65 y para la B, 66. Los códigos para las letras minúsculas a y b son 97 y 98. Cada letra minúscula tiene en el código ASCII un valor igual al valor de su equivalente en mayúsculas más 32. Esta equivalencia constante hace que resulte muy fácil convertir las series de caracteres de letras minúsculas en letras mayúsculas, y viceversa. Para realizar esta conversión necesitaríamos otras dos funciones que hasta ahora no habíamos utilizado nunca en nuestro curso de programación BASIC: ASC y CHR$.

La función ASC toma un carácter imprimible y lo devuelve con su equivalente en código ASCII, de modo que PRINT ASC("A") imprimiría en pantalla el número 65; PRINT ASC("b") imprimiría el 98.

La función CHR$ hace lo contrario: toma un número, da por sentado que está en código ASCII y devuelve el carácter que representa. De manera, entonces, que PRINT CHR$(65) imprimiría A, mientras que PRINT CHR$ (98) imprimiría b. Las funciones CHR$ y ASC se emplean mucho, junto con LEFT$, RIGHT$ y MID$, en los programas que se valen en gran medida de las series de caracteres. A continuación le ofrecemos un breve programa que acepta un carácter del teclado, verifica si es una mayúscula, y, si no lo es, lo convierte en mayúscula:

```
10 REM CONVERTIDOR DE MINUSCULA
   A MAYUSCULA
20 PRINT "DE ENTRADA A UN CARACTER"
30 INPUT C$
40 LET C = ASC(C$)
50 IF C > 90 THEN LET C = C -32
60 PRINT CHR$(C)
```

En futuros capítulos de nuestro curso de programación veremos más a fondo este tipo de manipulación de variables.

Por último, en este resumen, echemos una mirada a funciones de las que quizá carezca la versión de BASIC del usuario. Casi todas las versiones de este lenguaje permiten que el programador cree nuevas funciones, y éstas son casi tan fáciles de utilizar como las incorporadas. La sentencia DEF le señala al BASIC que se está definiendo una nueva función. A continuación le mostramos cómo definir una función para calcular el volumen de una esfera (la fórmula es $V = \frac{4}{3}\pi r^3$, donde r es el radio de la esfera y π (pi) es la constante aproximadamente igual a 3,14159):

```
10 REM FUNCION PARA CALCULAR EL VOLUMEN
   DE UNA ESFERA
20 DEF FNV(X) = 4 * 3.14159 * X * X * X/3
30 PRINT "DE ENTRADA AL RADIO DE LA ESFERA"
40 INPUT R
50 PRINT "EL VOLUMEN DE UNA ESFERA DE
   RADIO";R;"ES"
60 PRINT FNV(R)
70 END
```

Esta forma de definir una función es bastante directa, pero observemos la línea con todo detalle:

```
DEFine   identificador de función
     ↓   ↓
20 DEF FNV(X) = 4 * 3.14159 * X * X * X/3
         ↑  ↑
   FuNción  variable ficticia
```

Cuando se define la función, a las letras FN les sigue una letra identificadora (V, en el caso de la función de arriba) y luego a ésta le sigue una "variable ficticia". Esta variable ficticia también se debe utilizar en la definición de la función a la derecha del signo igual. Cuando la función se emplea en un programa, en el lugar de la variable ficticia de la definición se puede utilizar cualquier variable numérica.

En un punto posterior del programa anterior sería igualmente posible hacer uso de la función "volumen de una esfera" del siguiente modo:

```
 999 LET A = 66
1000 LET B = FNV(A)
1010 PRINT B
1020 LET C = 5
```

```
1030 LET D = B + FNV(C)
1040 PRINT D
1050 LET G = FNV(16)
1060 PRINT G
```

Algunas versiones de BASIC permiten utilizar variables múltiples en la función definida. En consecuencia, una función para hallar el promedio de dos números se podría escribir:

```
100 DEF FNP(B,C) = (B + C)/2
110 INPUT "DE ENTRADA A DOS NUMEROS";B,C
120 LET P = FNP(B,C): REM LA FUNCION
        'PROMEDIO'
130 PRINT "EL PROMEDIO DE";B;"Y";C;"ES";P
```

Observe que la línea 110 combina en una sentencia el equivalente de dos separadas. La mayoría de los BASIC imprimirán automáticamente las palabras encerradas entre comillas dobles a continuación de la sentencia INPUT, de modo que a tenor de dicha característica, esta línea equivale a:

```
110 PRINT "DE ENTRADA A DOS NUMEROS"
115 INPUT B,C
```

La línea 120 también consigue incluir el equivalente de dos sentencias en una línea empleando los dos puntos (:) como separador. Las sentencias que normalmente pertenecerían a líneas separadas se pueden escribir en una sola línea siempre y cuando cada sentencia "independiente" esté separada de la anterior por los dos puntos. En los programas largos esto puede contribuir a ahorrar espacio, pero no es conveniente abusar de su utilización, por una parte, porque dificulta bastante la lectura de los programas y, por otra, porque aumenta las posibilidades de error.

Ahora sí hemos tratado todos los puntos principales del lenguaje BASIC. En próximos capítulos del curso de programación BASIC estudiaremos el desarrollo y el diseño del programa, en lugar de detalles relativos al BASIC.

Respuestas a los "Ejercicios" de la página 197

Bugs
Se producirá un "ERROR FUERA DE DATOS", porque en la sentencia DATA de la línea 130 debería haber un total de 12 valores. En segundo lugar surgirá un error en la línea 100, cuando intenta direccionar un elemento A(4,1). La línea 100 debería decir:

```
100 PRINT A(X,Y)
```

Asignación de valores
Ahora exponemos una versión que llevaría a cabo las funciones requeridas. Es posible que su programa sea distinto

```
10  DIM A(8,13)
20  FOR F = 1 TO 7
30  FOR C = 1 TO 12
40  READ A(F,C)
50  NEXT C
60  NEXT F
70  REM SUMAR TOTALES
80  GOSUB 300
90  REM IMPRESION DATOS REQUERIDOS
100 GOSUB 200
110 PRINT "¿MAS DATOS?"
120 PRINT "SI(S) O NO(N)"
130 INPUT A$
140 IF A$ = "N" THEN GOTO 160
150 GOTO 100
160 END
200 PRINT "¿QUE MES?"
210 PRINT "1-PARA ENERO,"
220 PRINT "13 PARA TOTAL, ETC"
230 INPUT M
240 PRINT "¿QUE GASTO?"
250 PRINT "1-PARA GASOLINA"
260 PRINT "8-PARA TOTAL, ETC"
270 INPUT X
280 PRINT "EL VALOR ES ";A(X,M)
290 RETURN
300 FOR F = 1 TO 7
310 LET T = 0
320 FOR C = 1 TO 12
330 LET T = T + A (F,C)
340 NEXT C
350 LET A(F,13) = T
360 NEXT F
370 FOR C = 1 TO 13
380 LET T = 0
390 FOR F = 1 TO 7
400 LET T = T + A (F,C)
410 NEXT F
420 LET A (8,C) = T
430 NEXT C
440 RETURN
500 REM SUS DATOS SIGUEN AQUI
510 REM OCHENTA Y CUATRO VALORES
520 REM DATOS 3100, 2600, ETC
```

Complementos al BASIC



El Dragon-32 posee una versión de ASCII no estandarizada, que no admite caracteres en minúscula, de modo que el programa convertidor de minúsculas a mayúsculas no podrá realizar su función; inténtelo, a pesar de ello, y lea el manual del ordenador para recabar más información acerca del juego de caracteres del modelo.



En el Oric-1, el Vic-20, el Dragon-32 y el Commodore 64 sólo se puede incluir una variable en el interior de los paréntesis



En el Spectrum y el ZX81, reemplace:

ASC(A$) por CODE A$

y sustituya:

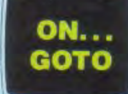
CHR$ (65) por CHR$ 65



Si el argumento es una expresión, se debe colocar entre paréntesis. Sin embargo, los argumentos simples (como A$ y 65) no necesitan paréntesis.



El ZX81, el Spectrum y el Lynx no disponen de estas sentencias.





En el BBC Micro sustituya

INPUT "CUALQUIER MENSAJE";M$$
por
INPUT "CUALQUIER MENSAJE",M$$



En el BBC Micro el usuario debe definir las funciones al final del programa, después de la palabra END o STOP, y no al comienzo del mismo, como en nuestro ejemplo. En este caso el BASIC del BBC es más eficaz que el BASIC estándar, de modo que para obtener mayor información consulte su manual.



En el Oric-1, en los dos fragmentos de programa que demuestran una buena estructura de bucle, reemplace:

IF INKEY$ = " " THEN LET X = 2
por
IF KEY$ = " " THEN LET X = 1

En el Dragon-32 sustitúyalo por:

IF INKEY$ = " " THEN LET X = 1

En el Vic-20 y en el Commodore 64 reemplace:

IF INKEY$ = " " THEN LET X = 2
por
GET A$: IF A$ = " " THEN LET X = 1

El acoplador acústico

Este dispositivo convierte datos digitales en tonos audibles y viceversa. Acoplado a un teléfono, permite comunicarse con otros ordenadores situados en cualquier lugar del planeta

Conectar su ordenador personal a una impresora o a un conjunto de unidades de disco es relativamente sencillo, porque generalmente éstos se hallarán en el mismo cuarto e, incluso, sobre la misma mesa. Pero conectar un ordenador con otros aparatos más grandes en su oficina o en cualquier otro lugar del mundo es ya una pretensión muy distinta. Por suerte, nosotros ya disponemos de un medio para la comunicación a través de todo el globo terráqueo: la red telefónica. Todo lo que se necesita es conectar el ordenador a este sistema.

Como la red telefónica es utilizada tan ampliamente por los circuitos de grandes ordenadores (los sistemas de venta de billetes para aviones de líneas comerciales, por ejemplo), la tecnología ya está bien establecida. Todo lo que necesita el usuario de un ordenador personal es una versión más barata y más sencilla de esta tecnología. El método convencional para conectar los ordenadores al sistema telefónico es un dispositivo denominado *modem*. Este extraño término no es más que un sencillo acrónimo formado a partir de las palabras modulador-demodulador.

El dispositivo funciona de forma muy parecida a la interface para cassette de un micro. Los patrones de unos y ceros binarios se convierten en señales eléctricas en dos frecuencias audibles diferentes y luego se envían a través de las líneas telefónicas (éste es el proceso de modulación). Al otro extremo, las frecuencias audibles se "demodulan" otra vez en unos y ceros. El modem envía una frecuencia constante (denominada *tono portador*), tanto si está enviando realmente información como si no, que hace que el ordenador receptor sepa que la línea aún está conectada.

El principal inconveniente de un modem es que ha de estar permanentemente acoplado por cable a la red telefónica, monopolizando su utilización, lo cual, en cierto sentido, representa una incomodidad para el usuario personal.

Un método de comunicación alternativo es el acoplador acústico. Dado que el sistema emplea tonos que se pueden oír, nada impide que éstos se puedan generar acústicamente, utilizando un altavoz. Éste se puede conectar luego al teléfono para que transmita a través del mismo. Al otro extremo, un micrófono colocado en contacto con el auricular del teléfono recogería la señal transmitida. Para esto está diseñado el acoplador acústico. A diferencia del modem, no necesita estar conectado permanentemente al teléfono.

Existen en el mercado diversos tipos de acoplador acústico, que van desde el dispositivo que muestra la ilustración, que es lo suficientemente compacto como para poderse conectar a un ordenador portátil, hasta unidades más grandes para escritorio. Las unidades sofisticadas se pueden utilizar para responder automáticamente a las llamadas que se reciben, sin necesidad de que esté presente el operador del ordenador, a través de la escucha constante del tono portador. Así como las interfaces para cassette varían en cuanto a la velocidad a la cual pueden almacenar y recuperar la información, la misma variación existe para los acopladores acústicos. No obstante, la gama de velocidades es estrictamente limitada. Las características de transmisión de un cable telefónico impiden que cualquier señal que supere los 1 200 caracteres por segundo (c.p.s.) pueda transmitirse con un nivel de fiabilidad razonable.

Las unidades económicas pueden funcionar a velocidades tan bajas como 30 c.p.s., mientras que los modelos más caros incorporan un interruptor para seleccionar una variedad de velocidades. Sin embargo, lo que hay que tener presente en todos los casos es que los dispositivos situados a ambos extremos de la línea telefónica siempre deben operar a la misma velocidad, ya que de lo contrario no se produciría la transmisión.

La utilización cada vez mayor de ordenadores personales en las gestiones económicas ha dado como resultado el desarrollo de gran cantidad de productos nuevos. Dispositivos como el Sendata y similares permiten que ordenadores portátiles como el Tandy 100 y el Epson HX-20 puedan ser utilizados como terminales de ordenador a distancia por cualquier persona, desde un periodista hasta un agente de ventas. Es posible darles entrada en la memoria del ordenador a los



Micrófono
Recoge la señal proveniente del altavoz del teléfono y alimenta con ella el tablero del circuito

Enchufe red
Le proporciona la alimentación eléctrica al acoplador desde un transformador adecuado. También se utiliza para recargar las pilas internas de niquel-cadmio

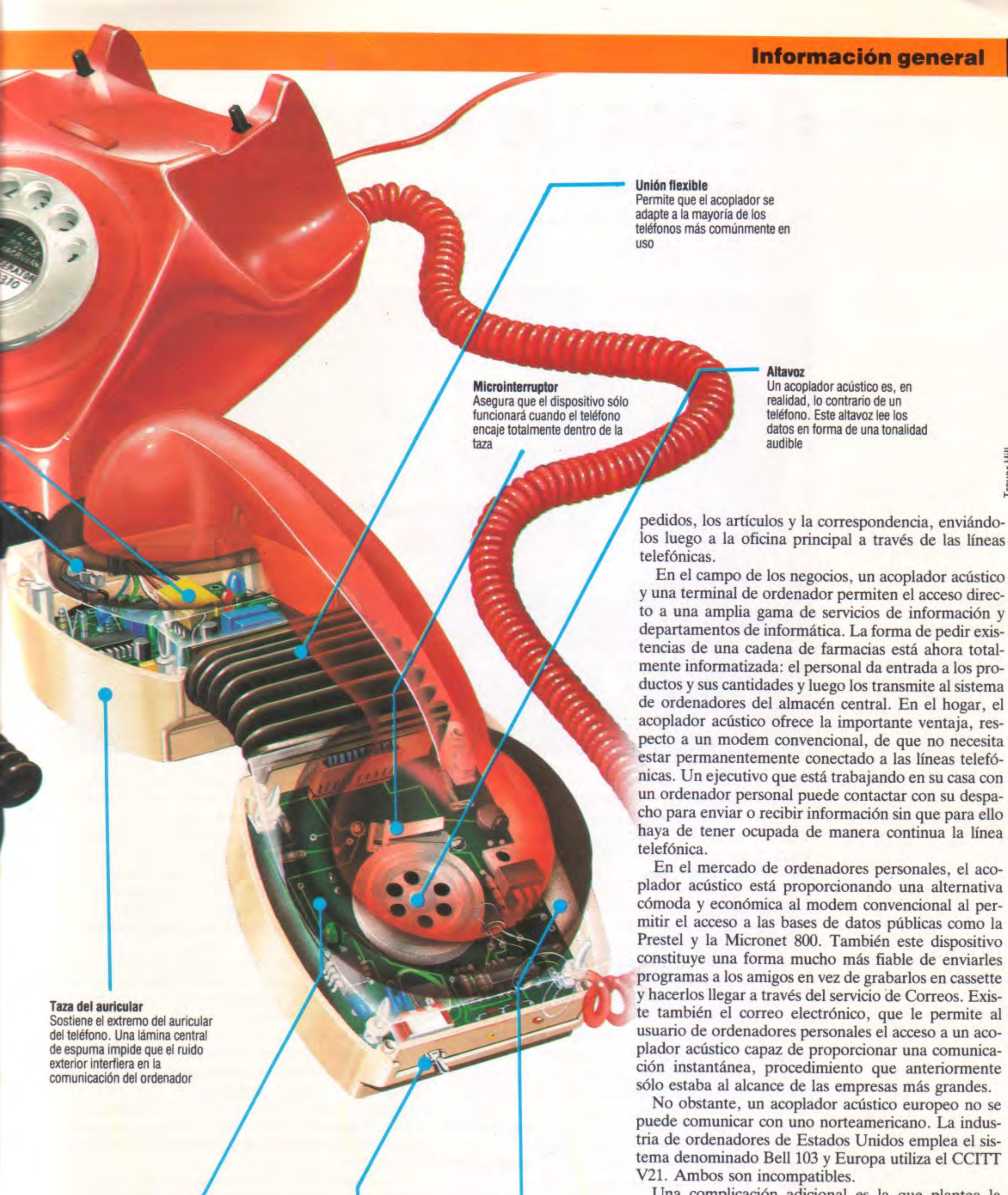


Cable para interface
Conecta con el enchufe para interface (en serie) RS232 del ordenador



Marcus Wilson-Smith

En una cabina telefónica
Los acopladores acústicos de poco peso permiten que el usuario de ordenador, estando de viaje, tenga acceso a la información de otros ordenadores, situados en cualquier lugar del mundo, a través de la red de teléfonos públicos



Trevor Hill

pedidos, los artículos y la correspondencia, enviándolos luego a la oficina principal a través de las líneas telefónicas.

En el campo de los negocios, un acoplador acústico y una terminal de ordenador permiten el acceso directo a una amplia gama de servicios de información y departamentos de informática. La forma de pedir existencias de una cadena de farmacias está ahora totalmente informatizada: el personal da entrada a los productos y sus cantidades y luego los transmite al sistema de ordenadores del almacén central. En el hogar, el acoplador acústico ofrece la importante ventaja, respecto a un modem convencional, de que no necesita estar permanentemente conectado a las líneas telefónicas. Un ejecutivo que está trabajando en su casa con un ordenador personal puede contactar con su despacho para enviar o recibir información sin que para ello haya de tener ocupada de manera continua la línea telefónica.

En el mercado de ordenadores personales, el acoplador acústico está proporcionando una alternativa cómoda y económica al modem convencional al permitir el acceso a las bases de datos públicas como la Prestel y la Micronet 800. También este dispositivo constituye una forma mucho más fiable de enviarles programas a los amigos en vez de grabarlos en cassette y hacerlos llegar a través del servicio de Correos. Existe también el correo electrónico, que le permite al usuario de ordenadores personales el acceso a un acoplador acústico capaz de proporcionar una comunicación instantánea, procedimiento que anteriormente sólo estaba al alcance de las empresas más grandes.

No obstante, un acoplador acústico europeo no se puede comunicar con uno norteamericano. La industria de ordenadores de Estados Unidos emplea el sistema denominado Bell 103 y Europa utiliza el CCITT V21. Ambos son incompatibles.

Una complicación adicional es la que plantea la nueva generación de teléfonos que se está introduciendo actualmente, y que parecen no adaptarse bien a la estructura (véase ilustración) de muchos acopladores acústicos. Dado que el sistema trabaja transmitiendo sonidos, es importante evitar que ruidos exteriores interfieran en la comunicación. Si el teléfono no se adapta correctamente o si hay muchos ruidos exteriores, existen muchas posibilidades de que los datos enviados se confundan.

Redes de información

Los centros de trabajo por ordenador de un mismo edificio se pueden enlazar entre sí con una red de área local. Así, varios usuarios comparten la información y los costosos periféricos



El uso de las redes se está generalizando en los centros de enseñanza, como el que muestra la fotografía, donde los jóvenes aprenden a utilizar los ordenadores y la microelectrónica

Tony Sleep

Ahora que se está haciendo familiar la presencia de los micros en las escuelas y oficinas, es cada vez más factible la posibilidad de que existan en un mismo edificio, e incluso en la misma dependencia, máquinas compatibles. En la medida en que este hecho se hace realidad, inevitablemente se tiende a pensar en los métodos que podrían utilizarse para vincular estas máquinas, aunque sólo fuera para compartir dispositivos periféricos como unidades de disco o impresoras. Los accesorios de esta clase son caros y compartirlos entre varios micros es una forma económica de emplearlos.

Pero la unión de periféricos no es, de ninguna manera, el único beneficio que se obtiene a partir del enlace de máquinas entre sí. Los micros se pueden montar a fin de que se comuniquen entre ellos formando una "red". Cuando todas las máquinas o centros de trabajo están situados en un solo edificio, se aplica el término *red de área local* (*Local Area Network*: LAN).

Se ha hablado mucho acerca de la posibilidad de abandonar el uso del papel en la oficina. Un primer paso hacia ello sería el de reemplazar el memorándum escrito por un mensaje que aparezca, desde una fuente lejana, en la pantalla de un monitor. La mayoría de las redes de área local, y también el Prestel, disponen de esta facilidad de "buzón". Para no distraer el trabajo que esté realizando en ese momento quien lo recibe, el mensaje recién llegado se anuncia en la línea inferior de la pantalla.

La aplicación de una red de área local puede ser muy útil en las escuelas. El texto se puede "reflejar" como en un espejo desde el micro del maestro a cualquiera o a todos los centros de trabajo de los estudiantes, o el maestro puede aprovechar esta instalación electrónica para revisar el trabajo de cada alumno, haciéndole comentarios y sugerencias.

La misma red de área local posibilita el empleo de una misma fuente común de información por usuarios diferentes. Tal vez la aplicación más frecuente de tal utilización se realice en las bases de datos que contienen información tanto pública como privada; el Prestel y los otros sistemas de videotexto constituyen ejemplo de ello.

Cuando un número determinado de personas está empleando el mismo archivo de información, es esencial asegurar que la información que éste contiene no se puede modificar sin que se les advierta a todos los usuarios. Por ejemplo, una fábrica puede utilizar una red de ordenadores para retener la información relativa a la disponibilidad de componentes y materias primas. Si a cada uno de los usuarios no se le presentara idéntica información, en el mejor de los casos no habría más que confusión y, en el peor, se le asignaría mercadería inexistente a quizá más de un departamento a la vez.

Al utilizar microordenadores con 64 Kbytes de RAM o más, resulta tentador "cargar", digamos, una sección de una base de datos y luego no volver a remitirse al archivo maestro hasta que se necesite examinar un segmento diferente. A menos que el software de control sea lo suficientemente amplio como para detectar y reflejar los cambios producidos en la información que contiene esa sección cargada, el usuario se podría estar refiriendo a cifras desfasadas.

El precio de gran parte del hardware de los ordenadores tiende a bajar, pero a menudo los periféricos

más avanzados técnicamente suelen costar más que el microordenador que los controla. Un buen ejemplo de esto es el Winchester, un disco rígido que funciona en un ambiente herméticamente cerrado (al vacío), y que ofrece parte de la velocidad y capacidad de almacenamiento de las unidades de disco principales. Los discos Winchester (así llamados algo arbitrariamente aludiendo al rifle de repetición Winchester 30/30, porque originalmente se construyeron como discos gemelos, cada uno de ellos con 30 megabytes de capacidad) son necesarios por el volumen de transacciones que maneja el software comercial. Sin embargo, tienen suficiente capacidad libre, aun en los emplazamientos de intensa actividad, como para hacer frente a más de un usuario de forma simultánea. Del mismo modo, mientras que una impresora matricial (véase p. 74) será adecuada para muchos trabajos, el tratamiento de textos suele requerir una impresora margarita, que suele costar más dinero. A menos que un solo usuario la pueda mantener en funcionamiento constante, para compensar su precio es conveniente que la impresora sea compartida por varios centros de trabajo.

Las redes ofrecen, asimismo, otras posibilidades. Los documentos que exigen la atención de varias personas se pueden traspasar de una a otra sin necesidad de imprimirlos en papel. Esto sucede, por ejemplo, en las redacciones de los periódicos y revistas técnicamente más adelantadas. El autor crea el material escrito a máquina, que luego entrega al redactor para que le dé su opinión crítica. Luego interviene el corrector, quien se encarga de revisar la ortografía, la gramática y el estilo del texto, marca las especificaciones técnicas y lo entrega al impresor, quien comienza el proceso de impresión. Antes de que existieran las redes de área local, las etapas por las que tenía que pasar el texto mecanografiado antes de llegar al impresor eran más diferenciadas y menos expeditas.

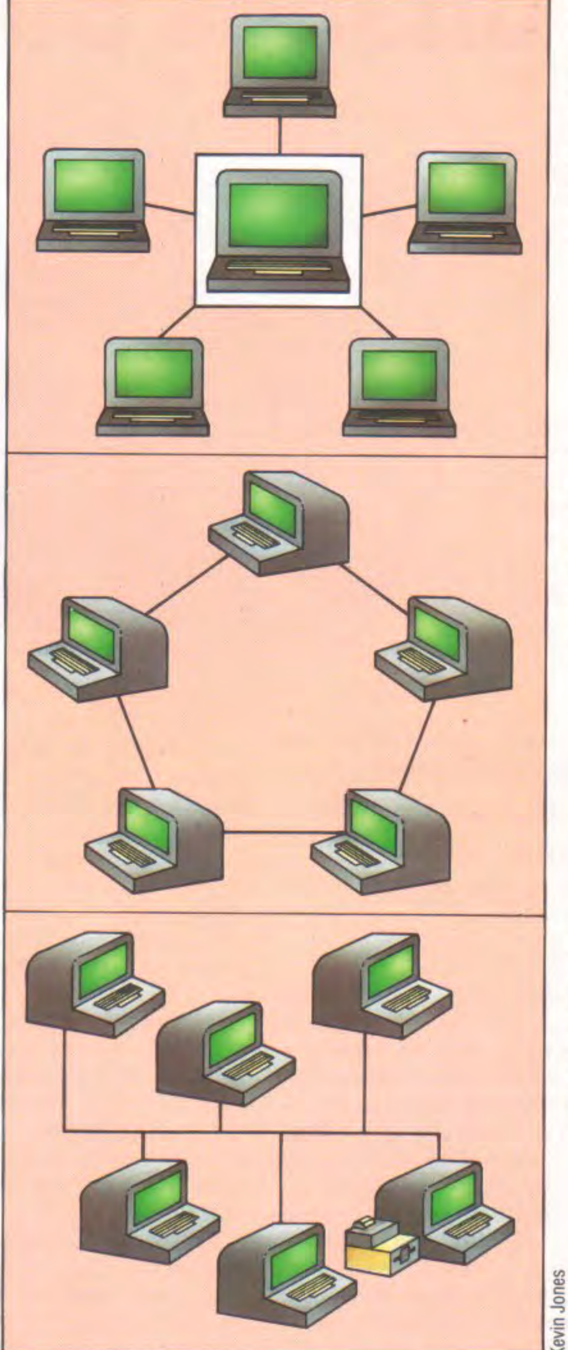
Los sistemas de red están a disposición de la mayoría de los ordenadores personales. Uno de los más utilizados en Gran Bretaña es el Econet de Acorn, concebido para el microordenador BBC. El Econet designa a una de las máquinas de la red para que actúe como "mensajero del archivo", que cuida de la unidad de disco central y se ocupa de las diversas solicitudes de información. Esta máquina puede estar dedicada exclusivamente a esa finalidad, o bien puede estar disponible para un usuario cualquiera siempre que no se la necesite para proporcionar un servicio a los otros miembros de la red. Si ésta comparte una impresora, se deberá disponer asimismo de una máquina destinada a controlarla.

El Econet puede abarcar hasta 254 centros de trabajo más los dos centros de "servicio", pero una limitación mucho más realista en cuanto a las dimensiones de la red viene dada por la mayor distancia a la cual puede estar situada la central más alejada respecto a la "unidad de reloj": un máximo de 500 metros. La unidad de reloj es una caja separada, también incluida en la red, que controla la velocidad a que se envía la información a través del sistema. El Econet utiliza dos pares de cables, como el sistema telefónico, y, al igual que éste, su instalación es relativamente sencilla. Uno de los pares transporta datos, y el otro los impulsos del reloj necesarios para asegurar la sincronización.

El software de comunicaciones del Econet, bastante sofisticado, reside en una EPROM (*Erasable Programmable Read-Only Memory:* memoria programable de lectura solamente, que puede borrarse) de ocho Kbytes situada en cada uno de los centros de trabajo. La tarea más complicada del sistema es la de evitar las "colisiones", es decir, asegurar que sólo un miembro de la red esté transmitiendo en un momento dado. Existen otras redes similares basadas en ordenadores personales, si bien por lo general no ofrecen en igual medida estas amplias configuraciones.

Las redes basadas en miniordenadores y ordenadores de unidad principal han alcanzado un nivel de utilización masivo en el curso de los diez últimos años, y no están restringidas a un solo país. Muchas líneas aéreas utilizan sistemas de reserva y venta de billetes que abarcan todo el globo, transmitiendo sus datos por las líneas telefónicas o por satélite.

Con la actual generalización de la televisión por cable, resulta razonable esperar que aumente la utilización de las redes, centrada probablemente en un concepto similar al del Micronet 800, un sistema basado en el Prestel, que permite que los programas se carguen al microordenador BBC a través de las líneas telefónicas.



Tipos de redes

Star
Una red star (en estrella) conecta a los usuarios de cada máquina con un controlador central, que también dirigirá a los periféricos que se utilicen en común

Ring
Algunas redes exigen que los usuarios de los ordenadores estén unidos entre sí en un bucle continuo. Estas redes están menos popularizadas porque los datos han de pasar a través de la mitad de las máquinas del anillo antes de que lleguen a su destino

Bus
El diseño de redes más refinadas, como por ejemplo la Econet, es muy similar, en cuanto a su concepto, a la arquitectura de un microordenador moderno: los datos y los mensajes de control pasan directamente de un usuario a otro



Charles Babbage



Aunque jamás se pudieron terminar, las máquinas de este matemático inglés fueron las precursoras del ordenador moderno

"¡Por amor de Dios, desearía que estos cálculos se hubiesen efectuado a vapor!", exclamó Charles Babbage mientras se afanaba con las tablas del Calendario Náutico. El siglo XIX había desarrollado la energía de vapor, pero la navegación marítima precisa continuaba siendo un problema. La posición de un navío se determinaba observando la luna y utilizando luego tablas matemáticas que con frecuencia eran inexactas.

Fue en 1812 cuando Babbage pensó por primera vez en construir una máquina, que él denominó ingenio diferencial, que pudiera efectuar los laboriosos cálculos que requerían las tablas náuticas. Hacia 1823 había completado un pequeño modelo y le solicitó al gobierno una subvención para poder construir una máquina que funcionara. El ministro de Hacienda le entregó 1 500 libras y Babbage se propuso crear una máquina que eliminara los errores mediante la impresión automática de los resultados de sus propios cálculos.

Babbage se entregó en cuerpo y alma a cumplir el objetivo fijado. El proyecto consumió enormes cantidades de dinero, pues sus expectativas se hallaban drásticamente limitadas por la insuficiencia de los conocimientos de ingeniería de aquel entonces. Obtuvo el dinero necesario gracias a la ayuda del primer ministro, su amigo el duque de Wellington. A pesar de la confianza de Babbage en que "lo que hiciera la máquina, lo haría con precisión", el gobierno decidió, finalmente, retirar su subvención al proyecto, luego de haber invertido 17 000 libras en él. El ingeniero que colaboraba con Babbage, Joseph Clement, también dimitió al poco tiempo, a raíz de una controversia, y se llevó consigo todas las herramientas que se habían diseñado específicamente para el ingenio.

Babbage se abocó rápidamente a un proyecto más ambicioso, el ingenio analítico, con el que esperaba alcanzar todos los objetivos para los cuales había construido el ingenio diferencial y muchos otros más aparte de ellos. En muchos sentidos su diseño se parecía al del ordenador moderno. Contenía un almacén de memoria y un "molino" aritmético (equivalente a la CPU), proporcionaba una salida impresa e incluso era posible programarla, mediante el empleo de bifurcaciones condicionadas.

Al principio las instrucciones se controlaban mediante clavos largos, como en un organillo; posteriormente se adoptó el sistema de tarjeta perforada que Joseph Jacquard había introducido en la industria textil. Babbage también experimentó con distintas bases numéricas pero, como todas sus máquinas eran mecánicas, la utilización del sistema binario no suponía ventaja alguna.

La compañera de Babbage, la condesa Ada Lovelace, matemática genial, se unió al proyecto. Ambos se hallaban abrumados por las dificultades, entre las cuales las económicas no eran las menores. Ella perdió gran parte de su patrimonio apostando en las carreras de caballos, aplicando a este juego un sistema "infalible", según ella. Después del fallecimiento de la condesa, acaecido cuando sólo contaba 36 años, Babbage continuó en solitario su labor.

Hombre de portentosa energía, también inventó el oftalmoscopio médico para examinar el fondo del ojo, hizo la coreografía de un ballet, ideó un sistema para iluminación del escenario e inventó una técnica para la señalización marítima.

En los últimos años de su vida se volvió irascible. Rechazó el título de barón que se le ofreció en reconocimiento de su trabajo, debido a que él aspiraba a que se lo nombrara par.

Con su trabajo Babbage anticipó la estructura del ordenador electrónico moderno, pero fracasó en convertir en realidad su visión global. Su ingenio analítico jamás llegó a terminarse, coartada su realización en razón de las limitaciones técnicas de la ingeniería del siglo XIX.

1792
Nace en Totnes, condado de Devon (Gran Bretaña), el 26 de diciembre

1810
Ingresa en el Trinity College, Cambridge, para estudiar matemáticas

1814
Se casa con Georgina Whitemore

1822
Publica un trabajo titulado *Observations on the Applications of Machinery to the Computation of Mathematical Tables (Observaciones sobre las aplicaciones de maquinaria al cálculo de tablas matemáticas).* Recibe la primera medalla de oro de la Astronomical Society, que él contribuyó a fundar

1827
Cambridge lo designa Lucasian Professor, cátedra que anteriormente ejerciera Newton, con un sueldo de 80 libras al año, aunque no resida allí ni imparta clase alguna

1833
Candidato al Parlamento por Finsbury

1834
Se suspende el trabajo sobre el ingenio diferencial después de la dimisión del ingeniero Joseph Clement

1862
El ingenio diferencial, parcialmente completo, se exhibe en South Kensington (Londres)

1871
Muere el 18 de octubre

Contra la máquina

Algunos de los microordenadores más potentes que existen hoy no están sólo en las oficinas o las fábricas de alta tecnología, sino también en las salas recreativas, los bares y las cafeterías



Marcus Wilson-Smith

Invasión extraterrestre
"Tienes que eliminarlo antes de que llegue al punto de explosión atómica, o se reproducirá clónicamente... ¡Cuidado! La nave madre se está preparando para sacar su cápsula... Será mejor que uses la Bomba Inteligente..."
No, no se trata de un diálogo perteneciente a una película de ciencia-ficción, sino de una conversación oída en una sala de juegos recreativos

En 1971, un joven llamado Nolan Bushnell invirtió gran cantidad de tiempo y de energía tratando de convencer a los propietarios de los bares y las cafeterías de los alrededores de Sunnyvale (California), donde vivía, para que accedieran a probar un nuevo tipo de juego que él mismo había inventado. Se trataba de una máquina tragamonedas, pero respondía a un concepto totalmente nuevo que ahora conocemos como televisión interactiva.

Finalmente el propietario de un bar accedió a probarla. Dos días después le telefoneó a Bushnell, quejándose porque la máquina se había averiado. Cuando Bushnell llegó al bar, descubrió que no había avería: la caja de las monedas estaba repleta.

El *Pong* (un derivado del tenis de mesa o ping-pong) fue el precursor de todos los emocionantes e innovadores juegos recreativos basados en microordenadores que ahora se pueden hallar en cualquier rincón del mundo. Tal vez sea interesante anotar que, si bien Bushnell fue el primero en sugerir que juegos como el tenis de mesa se podían simular mediante un ordenador, aun así no había llegado aún a formular el avance innovador decisivo: su juego exigía dos jugadores que compitieran entre sí, en vez de que fuera un solo jugador el que empleara toda su capacidad e inteligencia contra la máquina. Hubo de transcurrir un tiempo sorprendentemente largo hasta que surgiera la siguiente generación de juegos recreativos. No fue hasta 1977 que una empresa japonesa denominada Taito apareció en escena con el *Space invaders* (Invasores del espacio), que obtuvo un éxito enorme.

En la terminología que se emplea actualmente, *Space invaders* se conoce como un juego *alien zapping with shields* (eliminar extraterrestres, protegiéndose con escudos). El jugador desplaza su estación de disparo a lo largo de la parte inferior de la pantalla, protegiéndose con escudos cada vez que se siente amenazado, y disparando contra una línea de seres extraterrestres estilizados que siempre avanzan hacia él y que también le disparan a intervalos irregulares. El ritmo del avance de los extraterrestres es totalmente predecible y se acompaña con un ruido eléctrico apropiado en dos tonos que marca el paso de la desaceleración.

El *Space invaders* original utilizaba una visualización de televisión monocromática y gráficos de barrido. Era muy poco innovador en términos tanto de software como de hardware, pero cuando apareció por primera vez provocó una revolución social cuyo

precedente más cercano se remonta al nacimiento del cine. El hardware que soportaba al *Space invaders* y sus congéneres más cercanos apenas se diferenciaba del de los micros personales de entonces. Los fabricantes de estos ordenadores comprendieron rápidamente que existía un gran mercado aún por explotar en los prósperos hogares de Occidente. Las campañas de comercialización pasaron con prontitud de dirigir su enfoque hacia el aprendizaje de lenguajes para escribir programas para ordenador a centrarse en la utilización de éste como un medio de entretenimiento, y el diseño de las máquinas siguió el mismo rumbo.

Siete años después, los ordenadores aún se siguen vendiendo en función de su potencial como fuente de juegos, si bien actualmente la última generación de juegos recreativos ha ido mucho más allá, equiparando su capacidad a la de los ordenadores personales más avanzados. Es bastante frecuente encontrarse con un juego recreativo que tenga hasta un millón de bytes de memoria, así como una capacidad para gráficos muy raros de encontrar como no sea en los terminales para gráficos construidos especialmente para que los utilicen, con ordenadores de unidad principal, diseñadores, arquitectos, etc.



Muchos de los adelantos que se han producido en el campo de la informática personal se remontan a los juegos recreativos. El paso hacia los procesadores de 16 e incluso de 32 bits, por ejemplo, se produjo a consecuencia de los requerimientos de los usuarios de direccionar más de 64 Kbytes y de su necesidad de mayores velocidades de procesamiento. Los fabricantes de juegos recreativos estuvieron en la vanguardia de este movimiento y fueron unos de los primeros clientes de procesadores de 16 bits como el 68000 de Motorola y el 8086 de Intel. Sus exigencias en cuanto a un procesamiento más rápido y mayores memorias precedieron incluso a aquellas de los usuarios de microordenadores de oficina.

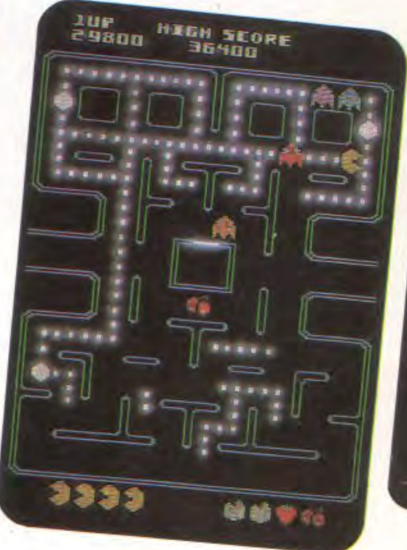
Todo ello ha sido de gran beneficio para el usuario de ordenadores personales, incluso de los del extremo inferior de la escala. Los gráficos sprite, por ejemplo, se desarrollaron para los juegos recreativos y posteriormente comenzaron a estar disponibles para los ordenadores personales. Chips especializados para gráficos y generación de sonido, como el Video Interface Chip y el Sound Interface Device, de Commodore, surgieron todos de la misma fuente, al igual que los tres chips que empleara Atari con la misma finalidad en sus ordenadores personales de las series 400 y 800.

Atari, cuyos éxitos se deben todos al juego original de Nolan Bushnell, constituye un ejemplo particularmente adecuado de la interrelación entre los juegos recreativos y los ordenadores personales. Hace mucho tiempo que Atari viene considerando el ordenador personal antes que nada como un medio de entretenimiento (un reflejo, quizá, del interés primordial de su propietario, ya que Atari pertenece a la Warner Bros) y, además de su gama de micros, también ofrece un ordenador exclusivo para juegos, el VCS (Video Cartridge System), que introduce en el hogar, virtualmente sin ninguna modificación, muchos de los juegos existentes para máquinas recreativas. En este sentido Atari tiene, por supuesto, la gran ventaja de que es una de las principales productoras de máquinas recreativas. Sin embargo, otras empresas dedicadas a cubrir el campo del entretenimiento y el ocio, en particular CBS (Columbia Broadcasting System) y la firma fabricante de juguetes Mattel, están también muy comprometidas en el mismo negocio.

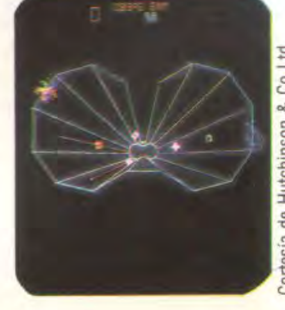
Todo el mercado del entretenimiento, trátese de juegos recreativos unitarios como de software de juegos para micros personales, se está convirtiendo rápidamente en toda una industria en sí mismo. En términos de estrategia de marketing guarda mucha semejanza con la industria discográfica, al proporcionar regularmente listas que incluyen los 20 juegos más vendidos. Ambos fenómenos ofrecen, asimismo, otros puntos de similitud. El software de juegos parece ser una de las obsesiones actuales de los niños en edad escolar y existe un floreciente negocio de piratería de software (véase p. 192).

Hemos mencionado antes que el nivel de desarrollo del hardware que se puede encontrar en los juegos recreativos los convierte en una clase diferente respecto a los micros personales. El más reciente de estos adelantos técnicos introduce la utilización de videodiscos para proporcionar, en el monitor de televisión, un telón de fondo sobre el cual se realizan los juegos. Si refinamientos como éste llegarán algún día al mercado personal, es tema de pura especulación. Ciertamente, existe en el campo de la electrónica tecnología avanzada a precio razonable para el consumidor, aunque el software de juegos que se produce para el mercado personal, tanto para consolas exclusivas para juegos como para micros personales, no ha llegado al nivel de los juegos recreativos en su forma original.

Cuando hablábamos de los simuladores de vuelo (véase p. 201), señalábamos que todos los juegos recreativos son simulacros de una u otra clase, ya sea de







Campos de juego
Los juegos recreativos como *Space invaders*, en los cuales el jugador puede mover su "señal" sólo a lo largo de una línea fija, han sido superados en popularidad por los juegos de "persecución en laberintos" como *Pac-Man*. Ambos utilizaban gráficos sprite y, en consecuencia, su calidad visual era pobre. Recientemente, los diseñadores de juegos han estado produciendo representaciones mucho más abstractas, como *Battlezone*, en que un jugador futurista combate contra un juego de tanques y misiles, o *Tempest*, que utiliza un asombroso diseño de gráficos para producir una magnífica ilusión de profundidad

Cortesía de Hutchinson & Co Ltd

Cápsula espacial

Astron belt es un ejemplo típico de la nueva generación de juegos recreativos que utilizan discos láser (véase abajo) para proporcionarle al juego un fondo en movimiento. Como los discos son dispositivos de acceso al azar, se puede pasar directamente de una escena a otra: una confrontación contra una nave extraterrestre que concluye con una explosión, por ejemplo. Para el jugador esto representa un gran paso hacia un mayor realismo. Las representaciones del disco láser pueden ser imágenes en movimiento de la vida real o bien imágenes animadas generadas por ordenador. El *Astron belt* también ofrece sonido estereofónico e incluso un asiento vibrador que reacciona a las bajas frecuencias del circuito de audio



Tony Lodge



situaciones de la vida real, como un juego de tenis de mesa o una carrera de coches, o bien una fantasía, como *Space invaders, Pac-Man* o *Frogger*. En los doce años transcurridos desde que existen los juegos recreativos, estas dos amplias bases se han ampliado, convirtiéndose en corrientes de desarrollo paralelas, si bien los juegos de fantasías espaciales son más comunes.

Hemos hablado anteriormente de los dos juegos originales (el *Pong* y *Space invaders*), pero vale la pena referirnos a su desarrollo. El primero de los juegos de bate y pelota que enfrentó al hombre contra la máquina fue el *Breakout* y todas sus variantes, en el cual el jugador arroja una pelota contra una pared de ladrillos. Cada ladrillo al ser golpeado desaparece de la pantalla, y el objetivo consiste en hacerlos desaparecer todos sin perder la pelota. A partir de este esquema tenemos los "simulacros" de golf, snooker-pool y pinball. En líneas generales, cuanta más atención preste el programador a reproducir las fuerzas existentes en el mundo real (por ejemplo, la gravedad, la resistencia del viento o de la tierra, un golpe impreciso), mejor será el juego.

Pero estos criterios no se aplican a los juegos de fantasía. En este caso, el jugador en realidad compite contra la persona que programó la máquina para que jugara al juego en cuestión, actuando enteramente según *sus* reglas, en un universo de *su* creación. Después de *Space invaders*, la etapa siguiente fue incluir distintas clases de extraterrestres que atacaran de diferente manera a intervalos al azar. Después de esto, fue necesario introducir movilidad en la "señal de pantalla" del jugador, lo que llevó a la creación de juegos tales como el *Defender*, considerado por los entendidos como el mejor de todos los de este tipo.

Los juegos de línea fija, como el *Space invaders* original, siguieron evolucionando y se convirtieron en lo que es el *Missile command* y sus derivados, cuyo objetivo consiste en defender la base propia contra el ataque de misiles intercontinentales. Los juegos de "persecución en laberintos", como *Pac-Man*, se desarrollaron a partir de los juegos de carreras de coches originales, donde el fin último consistía en conducir una señal en forma de coche por un recorrido la mayor cantidad de veces posible en el tiempo permitido, sin chocar contra las paredes. En estos juegos no había un verdadero elemento de competición (ni siquiera considerando las manchas de aceite que aparecían al azar como por arte de magia), de manera que el paso siguiente fue el convertir la "prueba de tiempo" en una persecución. La pista de coches original se convirtió entonces en un laberinto y las señales, por su parte, fueron sustituidas por frutas, bulbos luminosos y cosas por el estilo.

Los juegos de persecuciones de coches se orientaron hacia una representación seudotridimensional del recorrido, contemplado desde el interior del coche o desde detrás de él, con la carretera, siempre cambiante, abalanzándose sobre el jugador. Un método muy similar es el que se utiliza en los juegos de simulación de vuelo más realistas.

Por último, están los juegos tradicionales de tablero, como las damas, el ajedrez y el backgammon. Éstos están confinados a las aplicaciones para ordenadores personales, porque normalmente para jugar a ellos se necesita mucho más tiempo y la representación de gráficos ocupa un segundo lugar en importancia, después de los propios algoritmos de juegos del programa.

La única diferencia técnica verdadera radica en el método utilizado para generar los gráficos. La totalidad de los juegos más antiguos, así como la mayor parte de los actuales, emplean gráficos de barrido; pero algunos, en particular el *Asteroids*, utilizan métodos de exploración por vector, por lo cual el haz de electrones explora sólo las imágenes de la pantalla (no las zonas oscuras).

De modo que la próxima vez que pase por una sala recreativa repleta de juegos sumamente complicados, o que se apoye contra una de estas máquinas en un bar, recuerde que, en su interior, el ordenador que las acciona probablemente es mucho más complejo y poderoso que cualquiera de los que se emplean en el hogar o en una pequeña oficina, que utiliza muchas de las mismas técnicas y que su software ha sido creado por algunos de los mejores programadores que existen en la actualidad en todo el mundo.

Lo pequeño es bello

El microdisco de Sinclair es una respuesta a los problemas de costo y dimensiones que plantea el almacenamiento de datos. En vez de disco flexible usa un estrecho bucle de cinta magnética

Para el usuario de ordenadores personales, la cassette convencional representa un método barato y, por lo general, fiable para almacenar programas o cargar software comercial. Sin embargo, el sistema de cassette posee diversos inconvenientes. El problema principal es el que atañe a la velocidad; incluso un sistema de cassette rápido que funcione a 1 200 bits por segundo puede tardar varios minutos en cargar un programa largo o en buscar una información determinada. El segundo problema relevante se refiere a la cinta, que se mueve en un solo sentido: por lo general el ordenador no puede controlar los mandos de avance rápido y rebobinado. En el caso de que el programa esté almacenado en el extremo de una cinta, antes de empezar a cargarlo, será preciso bobinar la cinta entera a través de la grabadora.

Un sistema de disco resuelve todos estos inconvenientes, pero su precio es poco asequible. Lo que en realidad necesitan muchos usuarios de ordenadores

Interface 1
Además de proveer las conexiones necesarias para los microdiscos, esta unidad proporciona una conexión serial para acoplar impresoras y una interface en red que permite conectar entre sí hasta 64 ordenadores ZX Spectrum

Tambor accionador de cinta
La cinta es empujada a través del wafer por este tambor giratorio, que actúa como el tambor accionador de una grabadora de cassette

Cabeza de cinta
Se trata de una cabeza miniaturizada de grabación y reproducción, similar a las que poseen las grabadoras de cassette convencionales

Microinterruptor para protección de la cinta
Cuando se ha protegido el wafer quitando la lengüeta protectora de la información, se activa este microinterruptor, que evita que el microdisco grabe sobre el wafer

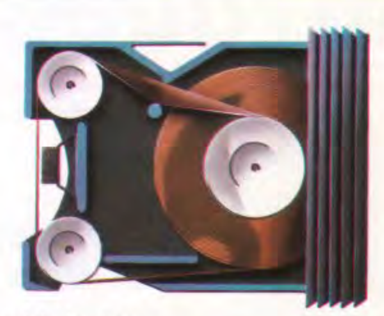
Conector para ampliación:
A través de este conector se pueden enchufar hasta ocho microdiscos

Microdisco cortesía de Sinclair Research

Indicador LED
Señala que se está utilizando el microdisco

Carga del wafer
Aquí se conecta el wafer del microdisco

personales es un dispositivo que sea mucho más veloz que una cassette y, a la vez, mucho más económico que un disco. Estos sistemas existen y se denominan *floppy tapes* (cintas flexibles). Desarrollada originalmente en Estados Unidos por Exactron para el sistema TRS-80 Modelo 1 de Tandy, la primera cinta flexible utilizaba un bucle de cinta continuo alojado en el



El wafer del microdisco
La cinta alojada en el interior del wafer del microdisco es un bucle de cinta de video continua de alrededor de 2 mm de ancho. Se utiliza cinta de video en vez de cinta de audio simplemente por su resistencia y su prolongada vida. Comparada con una cassette de audio, la cinta es más delgada y más estrecha, por lo cual es sumamente frágil.
En funcionamiento, el bucle de cinta circula dentro del microdisco en aproximadamente siete segundos y la información se transfiere a alrededor de seis Kbytes por segundo, un significativo avance respecto al 1,5 Kbyte por segundo de la interface para cassette del Spectrum. Cualquier programa almacenado en el wafer se puede hallar y cargar en aproximadamente 15 segundos. Cada cinta puede retener hasta 100 Kbytes de información, pero Sinclair sólo garantiza 86 Kbytes útiles. Cada wafer se ha de preparar para su utilización "dando formato" a la cinta, proceso que se activa mediante una sencilla orden en BASIC. A través del formato se verifica qué partes de la cinta se pueden emplear y se saltan los trozos en malas condiciones



Cable para interface
Este cable plano flexible conecta el primer microdisco con la unidad de interface

interior de un cartucho; la idea surgió del sistema de cinta de audio de ocho pistas que estaba de moda hace algunos años. El princpio de funcionamiento es sencillo: el bucle de cinta circula constantemente, de manera que los distintos programas se pueden hallar con muchísima más rapidez. También se lleva un catálogo de todos los programas y los archivos almacenados en la cinta (como el directorio de un disco), de manera que el usuario siempre tiene a su disposición una lista de contenido.

Como la información se graba digitalmente y no por métodos de audio, la transferencia de datos se puede efectuar con mucha mayor celeridad que con una cinta de cassette de audio: al menos cinco veces más rápido y, en ocasiones, aún más. No se requiere ninguna interface compleja ni cara: la unidad utiliza una conexión normal en paralelo y todo el software operativo necesario se encuentra incorporado en la unidad o bien se presenta como una ROM que se enchufa en un conector de reserva en el interior del ordenador. El nombre con que se conocen estas unidades describe su condición, a medio camino entre una cinta y un disco flexible: utilizan cinta pero funcionan como un disco.

Lamentablemente, las primeras unidades estuvieron plagadas de problemas. Los mecanismos funcionaban bien, pero las cintas resultaban poco fiables. El mayor fallo consistía en que la cinta, en realidad, no resistía la tensión de ser constantemente estirada desde una bobina y vuelta a bobinar en su exterior. Este problema no se presentaba en las cintas de audio de ocho pistas, que eran mucho más anchas y se movían mucho más lentamente. Hasta la introducción del microdisco de Sinclair, no es exagerado decir que el sistema de cinta flexible se consideraba algo así como una causa perdida.

El microdisco funciona exactamente según los mismos principios: un bucle de cinta pasa de manera constante a través de una cabeza de grabación y reproducción. Más aún: la cinta jamás había sido tan pequeña, con una anchura de alrededor de 2 mm (menos de la mitad que la cinta flexible original). Sin embargo, su fiabilidad está aún por demostrarse.

La cinta de cassette digital constituye un sistema de almacenamiento alternativo muy veloz y de probada fiabilidad. Hace ya mucho que las unidades profesionales están a la venta, pero a precios muy elevados. Con la introducción del sistema de microcassette digital de Philips han comenzado a aparecer dispositivos como la Hobbit (véase p. 94). A pesar de que la cinta no está en forma de bucle, ofrece una notable velocidad. El directorio se encuentra en la mitad de la cinta, que se puede bobinar en ambos sentidos bajo el control del sistema operativo. Recientemente se han incorporado aplicaciones de este tipo de microcassettes en los ordenadores portátiles Sharp PC-1251 y Epson HX-20.

El único problema importante de estos tres sistemas de almacenamiento es que el software disponible en sus respectivos formatos es limitado. Los ordenadores PC-1251 y HX-20 disponen de medios para cargar programas desde una cassette convencional y guardarlos después en la microcassette interior. La Hobbit y el microdisco se pueden conectar al ordenador simultáneamente como si se tratara de una grabadora de cassettes corriente, con lo cual se simplifica aún más el traspaso de información.

En términos de fiabilidad demostrada, el sistema de microcassette digital les lleva ventaja tanto al de cinta flexible como al de cassette convencional. La posibilidad de que con el tiempo llegue a reemplazarlos, entra en el campo de la especulación.

Sin rodeos

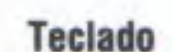
¿Qué ordenador adquirir y de qué modo mejorar su rendimiento con accesorios y periféricos? He aquí un informe objetivo sobre el hardware existente

Periféricos
Cuando se anuncia un ordenador nuevo, frecuentemente el folleto describe una gama de periféricos que tal vez no están ni siquiera en fase de diseño. Por lo tanto, es importante distinguir entre lo que se encuentra a la venta y lo que sólo está planificado para el futuro. El ordenador para el cual existan tanto unidad de disco como impresoras producidas por el mismo fabricante es una máquina para la cual se desarrollarán útiles programas de oficina. Para fines de entretenimiento, una máquina bien planificada habrá de contar con palancas de mando, mandos de raqueta y lápices ópticos disponibles, aunque en estos casos con frecuencia se puede sacar mejor partido de los accesorios suministrados por los proveedores independientes

> > > Spectrum, BBC Modelo B, Tandy Color, TI99/4A
> > Atari 400 y 800, Vic-20, Commodore 64, Dragon-32
> Oric-1, Sord M5

Pantalla
A los gráficos del ordenador se les presta mucha atención, cuando de hecho la mayor parte del tiempo es probable que el usuario esté mirando texto o listados de programas. Dos factores son importantes: el número de caracteres que se pueden visualizar simultáneamente (la cifra promedio es de 25 líneas de 40 caracteres) y la legibilidad de los caracteres. También son importantes las configuraciones de edición en pantalla. ¿Se pueden introducir alteraciones en un programa simplemente moviendo el cursor hasta la posición y digitando, o se deben utilizar órdenes especiales?

> > > BBC Modelo B, Atari 400 y 800, Commodore 64
> > Oric-1, Sord M5, TI99/4A
> Spectrum, Vic-20, Dragon-32, Tandy Color



Teclado
A menos que sólo pretenda jugar empleando una palanca de mando, utilizará su teclado con mucha frecuencia. Por lo tanto es importante que se sienta a gusto con él. La calidad de un teclado viene determinada, en primer lugar, por el diseño físico o el "tacto". Un mecanógrafo al tacto, por ejemplo, probablemente preferirá un teclado de los de mayor calidad, ya que algunos de los ordenadores más baratos están equipados con teclas transistorizadas que no ofrecen "feedback táctil". En segundo lugar está el trazado del teclado. El ideal es aquel que posee una tecla individual para cada función, ya que el hecho de que una misma tecla posea cierta cantidad de posibles aplicaciones diferentes puede llevar a confusión. Vale la pena disponer de teclas de función programable, porque se pueden utilizar en un programa para realizar funciones especiales como FIRE o START AGAIN

> > > BBC Modelo B, Atari 800, Vic-20, Commodore 64
> > Dragon-32, Tandy Color, TI99/4A
> Spectrum, Atari 400, Oric-1, Sord M5

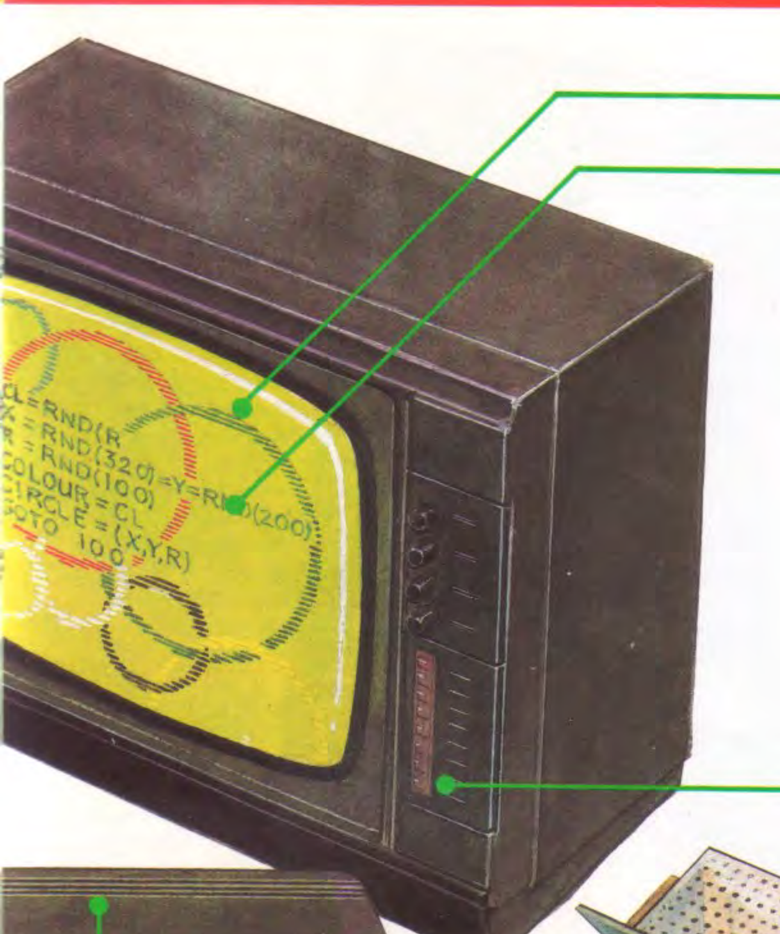
Programas de oficina
Si su razón principal para la adquisición de un microordenador reside en la capacidad de estas máquinas para llevar una pequeña empresa, debería pensar en un ordenador diseñado con ese fin. Sin embargo, los ordenadores personales más caros también se pueden utilizar como máquinas contables siempre y cuando esté dispuesto a invertir en una impresora y una unidad de disco. En ese caso las aplicaciones comprenden tratamiento de textos, contabilidad, bases de datos y hojas electrónicas.

> > > Atari 800, Commodore 64
> > BBC Modelo B, Atari 400, Vic-20, TI99/4A
> Spectrum, Dragon-32, Oric-1, Sord M5, Tandy Color

Interfaces
Si un ordenador posee una amplia gama de interfaces, es que ha sido diseñado previendo una futura ampliación, y esta característica constituye un excelente indicio de buen diseño en general. Disponer de conexiones estandarizadas como Centronics y RS232 significa poder utilizar una amplia gama de dispositivos para conectar a su ordenador, aparte de las unidades proporcionadas por el fabricante

> > > BBC Modelo B, Vic-20, Commodore 64
> > Atari 400 y 800, Dragon-32, Oric-1, Sord M5, Tandy Color
> Spectrum, TI99/4A





La calidad del BASIC
Casi todos los ordenadores personales vienen con un intérprete de BASIC incorporado en la ROM, pero como habrá observado a partir de nuestros recuadros de "Complementos al BASIC", las órdenes disponibles en cada uno de ellos varían considerablemente. Todas las versiones de BASIC que ostentan el nombre Microsoft trabajan aproximadamente de la misma manera, otras pueden utilizar estructuras completamente diferentes para series, matrices, etc. Un buen BASIC es aquel que posee muchas órdenes de "alto nivel", tales como CIRCLE, DRAW y PAINT

> > > BBC Modelo B, Dragon-32, Spectrum, Atari 400 y 800, Oric-1, Tandy Color, TI99/4A
> > Vic-20, Commodore 64
> Sord M5

Gráficos
Es difícil comparar la calidad de los gráficos que presentan los diversos ordenadores personales, ya que la mayoría de ellos ofrecen más de una modalidad. Digitar MODE 1, por ejemplo, podría darle acceso a 16 colores en una cuadrícula de 40 × 25 posiciones, mientras que MODE 7 podría proporcionar una resolución máxima de 320 × 200 pixels, pero con una elección de tan sólo dos colores. Debe tener en cuenta que en algunas máquinas la máxima resolución únicamente se puede apreciar si utiliza como pantalla un monitor en lugar de un televisor

> > > BBC Modelo B, Atari 400 y 800, Commodore 64, Sord M5, TI99/4A
> > Spectrum, Dragon-32, Oric-1, Tandy Color
> Vic-20

Sonido
Los ordenadores que utilizan el altavoz del televisor por lo general pueden producir un sonido de mayor calidad que los ordenadores que emplean un altavoz incorporado, aunque esto no es válido en todos los casos; de manera que es conveniente que antes de comprar un ordenador se solicite una demostración. Un sistema de voz única permite sólo notas únicas (como tocar el piano con un dedo), y tres o cuatro voces pueden realizar acordes. Los generadores de "ruido blanco" que poseen algunos ordenadores se utilizan para crear explosiones y otros efectos, mientras que aquellos con "regulador de envoltura" pueden imitar distintos instrumentos musicales, como un minisintetizador

> > > BBC Modelo B, Atari 400 y 800, Commodore 64, Oric-1, Sord M5, TI99/4A
> > Vic-20
> Spectrum, Dragon-32

Software de juegos
Gran parte de los mejores programas para juegos existentes no los vende el propio fabricante del ordenador sino proveedores especializados en software. Lo que determina tanto la calidad como la cantidad de los juegos producidos para cualquier máquina es el número de usuarios de ésta (y, en consecuencia, el volumen de ventas potenciales) y sus especificaciones técnicas. Por este motivo, algunos ordenadores parecen destinados a juegos mientras otros se orientan hacia la programación personal o de oficina

> > > Spectrum, Atari 400 y 800, Vic-20, TI99/4A
> > BBC Modelo B, Commodore 64, Dragon-32, Sord M5, Tandy Color
> Oric-1

Grabadora de cassette
A menos que esté dispuesto a ceñirse exclusivamente al cartucho de ROM o al software basado en disco, el usuario necesitará una grabadora de cassette. Muy pocos fabricantes exigen todavía que se compre su propia unidad; la mayoría de los ordenadores funcionan con cualquier grabadora de cassette doméstica, si bien tanto la velocidad como la fiabilidad con que se guardan y recuperan los programas varían mucho. Los mejores sistemas permiten controlar también el motor de la grabadora de cassette

> > > Spectrum, BBC Modelo B
> > Dragon-32, Oric-1, Sord M5, Tandy Color, TI99/4A
> Atari 400 y 800, Vic-20, Commodore 64

Memoria
Cuanta más RAM posee un ordenador, más sofisticados son los programas que puede ejecutar. Los ordenadores personales vienen con una memoria estándar que varía entre 1 y 64 K, susceptible de ampliación en la mayor parte de ellos. No obstante, el ordenador puede muy bien requerir parte de esta memoria para su propio uso interno (denominado *overheads* del sistema), dejando menor cantidad para el programa. La visualización de gráficos en color de alta resolución consume grandes cantidades de RAM

> > > Spectrum (48 K), BBC Modelo B, Atari 800, Commodore 64, Dragon-32, Oric-1 (48 K)
> > Atari 400, Tandy Color (16 K), TI99/4A
> Vic-20, Sord M5 (4 K)





Atención, compradores
Todos los meses aumenta en las tiendas la oferta de nuevos ordenadores, hecho que dificulta aún más la decisión del recién iniciado acerca de qué modelo elegir. Con tantos factores a considerar, la vieja advertencia de "cuidado, comprador" es más apropiada que nunca

Tanto si está pensando en adquirir su primer microordenador personal como si ya posee uno y desea ahora mejorarlo para conferirle un mayor nivel de sofisticación, las opciones disponibles pueden dejarlo bastante desconcertado. En *Mi Computer* hasta ahora hemos descrito con todo detalle algunos de los ordenadores personales que existen en el mercado, hemos explicado los principios operativos y los métodos operativos básicos, hemos hablado acerca de algunas aplicaciones de los microordenadores en nuestra vida cotidiana y hemos definido algunos términos básicos del especializado lenguaje de la informática.

Al comprar por primera vez un microordenador se han de tener en cuenta muchos factores y, entre todos ellos, quizá el primero sea el relativo al precio de la máquina. Ahora uno puede acudir a unos grandes almacenes y adquirir un Sinclair ZX81 en su forma más sencilla por menos de 20 000 pesetas. El principal inconveniente del ZX81 es la calidad o el "tacto" de su teclado. Con el fin de reducir al máximo posible sus dimensiones físicas, Sinclair adoptó para el teclado un diseño de membrana de capas múltiples. El resultado es una cierta falta de sensibilidad o de "realimentación táctil", que se puede superar al entrar en contacto permanente con la máquina. El especial diseño de este microordenador lo dota de dos características: de un lado, gran simplicidad; de otro, cierta lentitud.

No obstante, existen otras características a tener en cuenta por el futuro comprador, como la imposibilidad del Sinclair ZX81 para generar colores o producir sonidos y su poca capacidad para admitir accesorios.

Estas mismas consideraciones son válidas para muchos otros ordenadores económicos: por ejemplo, el Jupiter Ace o el Sharp PC-1500 (aunque tal vez el Ace sea interesante si el comprador tiene previsto llegar a utilizar el FORTH), y por lo tanto no los hemos tenido en cuenta a la hora de confeccionar este informe.

Existe toda una serie de características que uno esperaría hallar en un micro personal. Éstas se han detallado en la página anterior y resulta vital reseñarlas en una lista personalizada, por orden de importancia. Si, como les ocurre a muchos usuarios, considera que lo utilizará prioritariamente para juegos por ordenador, entonces le interesará una amplia base de software, periféricos apropiados, como palancas de mando o mandos de bola, buenos gráficos y capacidad para generar sonido. Alternativamente, si su interés se centra en el campo de la administración empresarial-doméstica, entonces tal vez para usted revista más importancia el número de las columnas visualizadas en pantalla, la calidad del teclado, el almacenamiento de la información, una conexión sencilla para impresora y gran cantidad de software.

Si bien los ordenadores son muy versátiles, algunos se prestan mejor que otros para una tarea determinada. Es verdaderamente esencial que confeccione una lista exhaustiva antes de salir a adquirir su primera máquina. Incluya en esa relación todas las características que espera hallar en su ordenador; quizá también sea conveniente asignar a cada punto un valor de uno a diez por orden de prioridad.

Para una máquina de uso general busque una buena versión de BASIC, un teclado cómodo, claras posibilidades de ampliación y una memoria estándar razonablemente grande.

Vale la pena mirar antes un poco. Es poco probable que consiga que le presten mucha atención en una cadena de grandes almacenes, pero es posible que consiga convencer al vendedor de que lo deje probar uno o dos juegos. No olvide el mercado de segunda mano. Miles de personas han comprado ya ordenadores personales. Muchas de ellas desean mejorar su potencial

informático hasta un punto en que resulta mucho más lógico adquirir un ordenador nuevo. Si bien las operaciones de compraventa de máquinas electrónicas no son un hecho totalmente desconocido, todavía distan mucho de ser algo corriente. Eche una ojeada a las páginas de "Se vende" de las publicaciones de informática y a las de su periódico local.

Cuando comience a pensar en mejorar las prestaciones de su ordenador o en comprar periféricos, piense que es muy importante estudiar el mercado con mucha atención. Si tiene un Spectrum, por ejemplo, se verá limitado a los propios periféricos de Sinclair que, a pesar de ser bastante eficaces, tienen un margen limitado de posibilidades. O es posible que su intención sea aumentar la capacidad de memoria de su Commodore Vic-20, entonces se encontrará con que debe optar por alguna de las propias unidades de Commodore o decantarse por otras muy parecidas producidas por fabricantes independientes, que o bien ofrecen lo mismo por menos dinero o quizá una potencia ligeramente inferior por igual precio.

Tal vez el área que ofrece mayores posibilidades de elección sea la de las impresoras. No sólo existe una gran cantidad de fabricantes de estas máquinas, sino que también se encuentran alrededor de media docena de diferentes tipos, siendo las más comunes las matriciales, las de rueda margarita y las impresoras de chorro de tinta. Si usted centra su interés en el tratamiento de textos, probablemente le convendrá una impresora de tipo de diseño margarita, que produce unos resultados comparables a las copias obtenidas con las mejores máquinas de escribir eléctricas. No obstante, si tiene pensado realizar presupuestos por medio de un programa de hoja electrónica, el primer punto que habrá de tener en cuenta es la máxima longitud de línea.

En términos de amplitud de elección, el segundo lugar es el del almacenamiento de información. ¿Es conveniente invertir en una de las más caras unidades de disco flexible, o, por el contrario, preferir una cinta flexible (véase p. 224)? Una vez más le diremos que mientras más dinero destine a la compra de su ordenador y demás dispositivos electrónicos, mayores posibilidades habrá de que sus demandas sean satisfechas.



Una de las cuestiones que más parecen preocupar al eventual comprador es: "¿Y si compro ahora y luego dentro de una semana el precio baja en un diez por ciento?" Todos los indicios hacen suponer que esta situación se prolongará en el futuro inmediato. Leyendo atinadamente las publicaciones de informática con frecuencia se pueden anticipar recortes de precio pero, en cualquier caso, ése no es un motivo válido para posponer la compra de un ordenador. Es mucho más importante que se esté seguro de que la máquina que se piensa adquirir es la adecuada, y, como hemos mencionado anteriormente, la forma más segura de acertar es pensar cuidadosamente en cuál será la aplicación principal a la que se la destinará. Recuerde siempre que su sistema informático es mucho más que la caja con teclado que se conecta con su aparato de televisión. Es más, incluso, que el software que haya adquirido y los periféricos. Su ordenador es una herramienta que, utilizada al máximo de su capacidad, puede proporcionarle entretenimiento, resolverle problemas y llevarle registros concisos y de fácil acceso.

Primero, el software
A menos que piense utilizar su ordenador sólo para aprender a programar, la disponibilidad de un buen software aplicativo será tan importante como las especificaciones de aquél, si no más. Cuanto más se venda un ordenador, mayores son las posibilidades de que se desarrolle software para él, y cuanto más software haya, mejor se venderá la máquina: una verdadera situación del tipo de si fue antes el huevo o la gallina



Ir de tiendas

Tiendas de descuento
Las tiendas que trabajan sobre la base de descuentos en equipos de alta fidelidad y videos suelen ofrecer microordenadores a buen precio



Grandes almacenes
Muchas personas compran sus ordenadores en los grandes almacenes, que los adquieren al por mayor. Al principio los vendedores de estos establecimientos tenían poco conocimiento acerca de estas máquinas; en la actualidad los grandes almacenes cuentan con departamentos de informática

Tienda especializada
Si le compra su ordenador a un comerciante profesional, puede estar seguro de contar con un eficiente servicio posventa y con consejos de absoluta confianza acerca del software, aunque es probable que no consiga el mejor precio





Venta por correo
Ésta solía ser una forma muy popular de vender microordenadores y, sin duda, el método más rentable para los fabricantes nuevos. Sin embargo, los largos períodos de espera para los nuevos ordenadores prometidos fueron la causa principal que determinó su pérdida de aceptación

Commodore Vic-20

La máquina más pequeña de la Commodore le ofrece al usuario unas configuraciones bastante sofisticadas a precio razonable

Commodore Business Machines (CBM) fue la responsable de uno de los primeros micros personales (el Personal Electronic Transactor: PET), que salió al mercado en 1977. En 1981 lanzó el Commodore Vic-20, que incorpora muchas características del PET. El Vic no sólo utiliza el mismo microprocesador 6502, sino incluso el mismo BASIC en ROM que, lamentablemente, presenta el inconveniente de que no se trata de la versión más eficaz ni la más reciente de las producidas por Commodore.

La diferencia más evidente entre las dos máquinas concierne a las capacidades para gráficos adicionales del Vic. Su nombre proviene del chip exclusivo que activa su visualización en pantalla: el Video Interface Chip. Ofrece una disponibilidad de hasta 16 colores, aunque la visualización consta de un marco o reborde para el cual hay ocho colores disponibles; un fondo, que puede ser de cualquiera de los 16 colores; y los caracteres o símbolos individuales, cuyos colores se pueden escoger entre un grupo de ocho.

El juego de caracteres en sí mismo es asombrosamente grande, ofreciendo tanto mayúsculas como minúsculas y dos juegos de caracteres para gráficos a partir de 62 teclas, además de cuatro teclas exclusivas que se pueden utilizar, pulsando SHIFT, para proporcionar ocho funciones programables. El diseño del teclado es particularmente acertado, tanto ergonómica como técnicamente.

El principal inconveniente del Vic es su pequeña capacidad de memoria: sólo cinco Kbytes, que se reducen a 3,5 Kbytes después de que el sistema operativo ha utilizado RAM para la pantalla y para otras exigencias internas. Sin embargo, tiene la ventaja de que es posible direccionar hasta 32 Kbytes de memoria, además de lo cual se puede disponer de memoria adicional desde diversas fuentes.

Se proporcionan conexiones para interfaces para mandos de raqueta, palanca de mando, lápiz óptico, cartuchos de juegos, ampliación de memoria, impresora, unidad de disco, cassette y televisor, y hay una que responde al estándar en serie RS232, que se puede emplear con un modem o con una impresora que no sea Commodore. Además de todo ello, existe una amplia gama de accesorios de hardware, los mismos que para el más reciente Commodore 64 (véase la sección "Modelos de Hardware" de página 49).

Conexión cassette
El Vic-20, al igual que otros micros Commodore, necesita de una grabadora de cassette fabricada especialmente, que se enchufa aquí

Conexión dispositivos
Este conector de 24 patillas es en serie y se utiliza para activar ciertos dispositivos periféricos adicionales

Adaptador para interface de periféricos
Estos chips controlan todas las operaciones de input/output del Vic-20, y poseen capacidad de procesamiento propia. Pueden, por ejemplo, efectuar la conversión entre formatos en serie y en paralelo

Conexión teclado
Aquí se conecta el teclado con el adaptador para interface de periféricos

Chip para interface de video
Este dispositivo incorporado especial se utiliza para controlar el estado de la pantalla y también el generador de sonido de tres voces del Vic-20

Memoria de acceso directo (RAM)
El Vic-20 está dotado de una RAM de 5 K, la cual se puede ampliar externamente a 32 K



Los periféricos del Vic
Commodore suministra una unidad de cassette, una unidad de disco y una gama de impresoras para el Vic-20, así como palancas de mando. Alternativamente, hay una gran variedad de dispositivos de otros fabricantes. De éstos, tal vez los más importantes sean los módulos para ampliación de memoria



Teclado
Las 62 teclas del teclado proporcionan cuatro juegos completos de caracteres, y permiten la utilización de hasta ocho funciones programables en virtud de las teclas exclusivas situadas a la derecha

Chris Stevens

Carga de cartuchos
Esta ranura acepta cartuchos de software ROM o placas de ampliación de memoria

Enchufe red
Aquí se conecta la unidad de alimentación eléctrica externa

Interruptor

Conexión juegos
Aquí se pueden conectar lápices ópticos, palancas de mando o mandos de raqueta

Enchufe audio-video
Aquí se conecta un modulador de RF externo

Conexión en serie
La mayoría de los periféricos del Vic, la impresora y la unidad de disco, por ejemplo, se accionan desde esta conexión diseñada especialmente que, a diferencia de la mayoría de las interfaces en serie, puede direccionar más de un dispositivo periférico a la vez

Microprocesador
El Vic-20 utiliza el microprocesador 6502 de MOS Technology, de probada eficacia y enorme popularidad

Memoria de lectura solamente (ROM)
Estos chips contienen el BASIC, el juego de caracteres y toda la información necesaria para interpretar y ejecutar los programas

Commodore Vic-20

DIMENSIONES
404 × 216 × 75 mm

PESO
1 820 g

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
La máquina viene con 5 K de memoria estándar ampliables a 32 K por medio de un adaptador de 16 K, otro de 8 K y un tercero de 3 K

VISUALIZACION EN VIDEO
23 líneas de 22 caracteres. Los gráficos, de alta resolución, permiten 184 × 175 pixels. Ofrece un máximo de 16 colores

INTERFACES
Dos conexiones en serie, enchufe audio-video, conexión cassette, carga de cartuchos, conexión juegos

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
Ensamblador y órdenes adicionales en BASIC

VIENE CON
Unidad de alimentación eléctrica, cable para antena, manual

TECLADO
Teclado de tamaño normal similar al de una máquina de escribir, con 62 teclas y otras cuatro de función especial

DOCUMENTACION
Una de las áreas menos cuidadas de la CBM es la de la documentación que ofrece con sus ordenadores personales. Los sistemas de gestión como el 8032 vienen con un ejemplar de la excelente obra de Adam Osborne "Guide to CBM Computing", pero a los propietarios del Vic se los relega a un plano mucho más trivial. Aunque está escrito con sencillez y, por consiguiente, resulta fácil de comprender, el manual del usuario apenas si se queda en la superficie de las configuraciones de la máquina. Afortunadamente, se encuentran a la venta gran cantidad de publicaciones que pueden cumplir un papel alternativo

Consultando la agenda

Valiéndonos de todas las técnicas de la programación en BASIC que hemos aprendido hasta el momento, damos ahora los primeros pasos para desarrollar un programa de base de datos

Ahora que ya hemos cubierto muchos de los fundamentos del BASIC, es el momento de poner en práctica cuanto hemos aprendido desarrollando un verdadero programa. Por supuesto, todos los programas con los cuales nos habíamos encontrado hasta ahora eran también programas "verdaderos" en el sentido de que realizaban una tarea específica, pero eran ilustraciones de cómo funcionan los diversos componentes del lenguaje BASIC y no la clase de programas que el usuario desearía emplear todos los días. Ilustraban cómo los "engranajes" del BASIC se podían unir entre sí para formar un mecanismo sencillo. En este momento estamos en disposición de ensamblar esos mecanismos. ¡Ahora es ya un reloj completo lo que vamos a construir!

Una de las preguntas más comunes que las personas que no emplean ordenadores les formulan a los usuarios de ordenadores es: "¿Para qué se puede utilizar realmente un ordenador?" La pregunta no es tan simplista como parecería a primera vista. Las respuestas más convencionales tienden a seguir la línea de: "Bueno, puedes informatizar tus recetas de cocina" o "Puedes crear una agenda de direcciones o de teléfonos informatizada". Es muy raro que esta táctica dé buenos resultados, porque quien hizo la pregunta por lo general hace comentarios como: "Pero yo puedo consultar mi libro de cocina cada vez que deseo cocinar algo y puedo leer mi agenda de direcciones cuando quiero hallar la dirección de alguien, sin necesidad de tener que pasarme horas escribiendo un programa para hacerlo". ¿Qué circunstancias se deben considerar para decidir si a un problema se le debe buscar respuesta por medio de la informática y no por métodos convencionales? Responderemos a esta pregunta trabajando en el ejemplo específico de una agenda de direcciones informatizada.

Una agenda de direcciones corriente comúnmente consta de un índice alfabético que se puede manipular con los dedos, diseñado para que el usuario pueda localizar, muy aproximadamente, un nombre determinado. Por lo general, y cuando es necesario, se suelen ir agregando otros nombres, y no por orden estrictamente alfabético. Su primera entrada en la página de la P podría ser Daniel Puig. Después se podría agregar Alfonso Prado o Camilo Pérez. Aunque estos nombres no está ordenados alfabéticamente, están agrupados juntos bajo la P, de modo que la tarea de hallar un apellido determinado que comience con P no será demasiado complicada. Por otra parte, si no se utilizara ningún tipo de índice, localizar un nombre sería un verdadero problema.

Las otras entradas corrientes en una agenda de direcciones son las señas y el número de teléfono de la persona e, incluso, alguna información personal. Sin embargo, una agenda de direcciones convencional no le puede dar una lista separada de todas las personas que viven en Barcelona, o a quién le corresponde un determinado número de teléfono. Puede que esto no represente un inconveniente grave, pero si usted fuera propietario de una pequeña empresa de ventas por correo sería una ayuda muy valiosa que pudiera obtener información específica acerca de las personas de su cartera de clientes. Por ejemplo, si quisiera distribuir una nueva línea de pijamas para niños podría anticipar nuevos pedidos informando a sus clientes, pero para ahorrarse algún dinero en sellos de Correo probablemente no valdría la pena enviarles los folletos a los clientes que no tuvieran hijos. Éste es el tipo de consideraciones que se deben evaluar antes de decidir si, frente a un problema dado, es más conveniente una solución informática o una solución convencional.

Si la solución informática es apropiada, la siguiente consideración ha de ser, entonces, si comprar o no un software existente a nivel comercial. Una mirada a los anuncios de las revistas de informática sugeriría que los programadores de ordenadores ya han pensado en todas las eventualidades posibles. Sin embargo, un examen más detenido podría dejar en evidencia el hecho de que un programa disponible comercialmente podría no ajustarse con toda exactitud a lo que se desea, o que no exista para su modelo de ordenador, o que sea demasiado caro. El costo de un programa refleja generalmente los costos de desarrollo. Un paquete de tratamiento de textos puede ser caro, pero si el usuario decidiera escribirlo él mismo y tuviera que destinar seis meses de dedicación exclusiva a su realización, sin duda le resultaría mucho más caro.

Desde el punto de vista de lo positivo, el software que se escriba usted mismo podría hacer exactamente lo que deseara que hiciera. El otro factor es la incomparable satisfacción de que, solo y sin ayuda de nadie, escriba usted un programa con el que obtenga un éxito rotundo.

El diseño de un programa consta de varias fases, la primera de las cuales es la comprensión profunda del problema, premisa que implica una descripción clara de dicho problema.

La segunda consiste en hallar un enfoque para solucionar el problema. Esto implica una descripción de la forma esperada de la entrada y la salida como un "primer nivel de descripción" del problema. Los problemas y las soluciones se deben plantear en los términos más amplios y éstos se han de concretar gradualmente hasta llegar a la etapa en la cual los podemos codificar en un lenguaje determinado.

La tercera etapa es la codificación en sí misma. Utilizaremos el BASIC como nuestro lenguaje de alto nivel, pero en lugar de éste podríamos emplear igualmente cualquier otro lenguaje. Hasta la etapa final de codificación en BASIC utilizaremos un seudolenguaje intermedio entre la libertad y flexibilidad del lenguaje corriente y las estructuras rígidas de un verdadero lenguaje para ordenador como el BASIC.

El enfoque a la programación que acabamos de des-

cribir se denomina generalmente programación *top-down* (de arriba/abajo). Se trabaja desde el nivel superior (una enunciación general de los objetivos globales), a través de varios niveles de refinamiento, hacia abajo, hasta los detalles más precisos del programa necesarios para empezar a codificar en el lenguaje de alto nivel escogido. También intentaremos ceñirnos a los principios de la denominada "programación estructurada". Estos principios se irán clarificando a través del desarrollo de este proyecto.

Los pasos que seguiremos en el desarrollo del programa se pueden resumir de la siguiente manera:

1. Una clara enunciación del problema
2. La forma de la entrada y la salida (primer nivel de descripción)
   2.1 Refinamiento (segundo nivel de descripción)
   2.2 Sucesivo refinamiento (del tercer al enésimo nivel de descripción)
3. Codificación en lenguaje de alto nivel

Antes de comprometernos en un proyecto importante de software, es esencial plantear el problema con toda claridad. Se trata de un ejercicio que no es en absoluto trivial. Probemos con algunas ideas para nuestra agenda de direcciones informatizada.

Empezaremos primero por una lista de las características deseables; luego podemos decidir cuáles de ellas se pueden emplear con un esfuerzo de programación razonable. Deseamos que se pueda:

1. Localizar una dirección, un número de teléfono y notas dando entrada a un nombre desde el teclado
2. Obtener una lista de nombres, direcciones y números de teléfono dando entrada a sólo parte de un nombre (sólo el primer apellido o el nombre)
3. Obtener una lista de nombres, direcciones y números de teléfono de una ciudad o unas zonas determinadas
4. Obtener un listado de todos los nombres que comienzan por una letra determinada
5. Obtener un listado completo de todos los nombres de la agenda de direcciones, ordenados alfabéticamente
6. Agregar todas las entradas nuevas que deseemos
7. Modificar la información ya existente
8. Suprimir los datos que deseemos

Supongamos que ya hemos escrito el programa para la agenda de direcciones. ¿Qué forma deberían asumir la entrada y la salida? ¿Cómo le gustaría que funcionara el programa desde el punto de vista del usuario? En términos generales, los programas pueden ser "activados por menú", "activados por órdenes" o por una combinación de ambos. En un programa activado por menú, cada vez que se ha de tomar una decisión se le ofrece al usuario una lista (menú) de opciones. La selección se suele efectuar pulsando sólo una tecla. En los programas activados por órdenes, el usuario digita las palabras-órdenes o frases-órdenes específicas, por lo general sin que se le requiera en ese sentido. Algunos programas combinan ambas técnicas. La ventaja de un programa activado por menú es que a un recién iniciado le debería resultar sencillo de utilizar, logrando que el programa fuera más "amable con el usuario". Un programa activado por órdenes debería resultarle más expedito al usuario experimentado. Nosotros optaremos por un enfoque activado por menú, si bien usted está en libertad de decidirse por ejecutar este programa utilizando, en cambio, rutinas de órdenes: la decisión es suya.

Dado que el programa se centrará en una lista de nombres, lo primero que debemos considerar es qué forma deben asumir esos nombres. ¿Comprenderá el ordenador, por ejemplo, todos los formatos siguientes?:

A. J. P. Soler
Leonardo da Vinci
Bo DEREK
Juana R.
L. Prado
j. j. García
F López
Twiggy
GROUCHO MARX
Sir Freddie Smith

Esto puede sonar como que estamos hilando muy fino, pero consideremos lo que sucedería si diera entrada a `F López` y después le pidiera al programa que buscara `F. López`. A menos que usted hubiera previsto el problema, probablemente el ordenador le respondería `NOMBRE NO HALLADO`.

Existen dos formas de afrontar el problema: podemos tener una entrada "libre", que permita dar entrada a los nombres en cualquier forma junto con rutinas inteligentes que tomen en consideración este hecho cuando se realiza la búsqueda; o bien podemos insistir en que se dé entrada a los nombres en una forma estrictamente definida. Todo nombre que no se ciñera a ésta produciría un mensaje de error como `FORMATO DE NOMBRE INACEPTABLE`. La elección es de índole arbitraria, pero tenemos la posibilidad de inclinarnos por la opción de una entrada muy "libre" y dejar que el programa se preocupe de convertir los nombres en una forma estándar.

Desde el punto de vista de una búsqueda alfabética, podemos imaginar que los nombres poseen dos partes: el nombre de pila y el primer apellido. Un apellido es relativamente sencillo de definir: una serie de caracteres alfabéticos en mayúsculas o minúsculas terminados en un Carriage Return (retorno de carro) y precedidos por un espacio (ASCII 32). De inmediato se presenta un problema: ¿qué sucedería si se diera entrada sólo al nombre "Twggy" sin estar precedido por un espacio? Presumiblemente el programa lo rechazaría por tener un formato inaceptable. Será mejor que modifiquemos nuestra definición.

Consideremos que un nombre consta de una o dos partes: el apellido o el apellido y el nombre de pila. El nombre puede incluir tanto caracteres en mayúsculas o en minúsculas, como puntos, apóstrofos y guiones. Siempre comenzará con un carácter alfabético y terminará con un "Carriage Return" (no se admitirán puntos al terminar). De haber un espacio, el último grupo de caracteres (incluyendo apóstrofos y guiones) se contará como el apellido y otras partes, incluyendo el espacio, se contarán como el nombre de pila. De no haber espacio, el nombre completo se considerará como el apellido.

El apellido requiere una consideración especial porque, en el ordenamiento alfabético, precede siempre a los nombres de pila. De manera que Ana Pérez vendría después de Victoria Paredes. Si un nombre se compone de solo un grupo de caracteres, como Trevanian, Twiggy o un apodo como "Gordo", a los efectos de nuestro programa se lo puede considerar como un apellido.

En una búsqueda alfabética, ¿qué nombre iría primero: A.J.P. Soler o Alfredo Soler? La decisión es arbitraria, pero la solución más sencilla sería ignorar los caracteres no alfabéticos comprendidos antes del último espacio y hacer que los nombres equivalgan a AJP Soler y ALFREDO SOLER. Al proceder de esta manera, tanto ALFREDO como AJP se considerarían como nombres de pila, y AJP iría primero.

Parte de nuestro programa aceptaría como entrada un nombre y produciría como salida un nombre, una dirección y un número de teléfono (observe que aún no hemos ni siquiera comenzado a considerar los significados de "dirección" y "número de teléfono"). Si aceptáramos como entrada nombres de formato "libre", con conversión interna a un formato estandarizado, ¿esperaríamos que la salida estuviera en la forma "estandarizada" o en la misma forma que la entrada original? La salida más cómoda para el usuario sería aquella en la que el nombre estuviera en la forma original, pero, como veremos, esto complicará la programación.

Como tarea inicial de programación, supongamos que se le ha asignado un nombre a la variable alfanumérica NOMBRE$ (nombre completo) y que tenemos otras dos variables, NOMB$ (nombre de pila) y APELL$ (apellido). ¿Cómo le asignaremos a NOMB$ y a APELL$ las partes apropiadas de NOMBRES? Ignorando, de momento, el problema de llevar un registro de la forma original en que se ha dado entrada al nombre (de modo que se pueda recuperar cuando la necesitemos más adelante), una sentencia simple del programa podría ser:

Convertir en mayúsculas todos los caracteres
Eliminar todos los caracteres no alfabéticos excepto el espacio final
Asignar a APELL$ todos los caracteres que siguen al último espacio
Asignar a NOMB$ todos los caracteres que preceden al último espacio

Antes de considerar cómo se podría codificar este problema en BASIC, veremos cómo el proceso de "programación de arriba abajo" nos puede llevar desde una enunciación muy amplia de nuestro objetivo hasta el punto en el que se hace posible la codificación en un lenguaje de programación determinado. Observará que estamos utilizando no sólo nombres de variables largos como NOMBRES, sino palabras-órdenes como BEGIN, LOOP y ENDLOOP. Éstas son construcciones que hemos inventado para que nos ayuden a describir nuestro programa. En la etapa final de desarrollo serán reemplazadas por órdenes equivalentes en BASIC. Explicaremos mejor estas órdenes, así como por qué hemos sangrado algunas de las líneas, en el próximo capítulo de nuestra obra.

1.ª ENUNCIACION DE OBJETIVOS

```
INPUT
Un nombre (en cualquier formato)
OUTPUT
1. El nombre de pila
2. El apellido
```

1.er REFINAMIENTO

```
1. Leer NOMBRE$
2. Convertir en mayúsculas todas las letras
3. Hallar último espacio
4. Leer APELL$
5. Leer NOMB$
6. Eliminar caracteres no alfabéticos de NOMB$
```

2.º REFINAMIENTO

```
1. Leer NOMBRE$
2. (Convertir en mayúsculas todas las letras)
   BEGIN
   LOOP mientras los caracteres no explorados
   permanecen en NOMBRE$
      Leer los caracteres de NOMBRE$ sucesivamente
      IF el carácter está en minúscula
         THEN convertir en mayúscula
         ELSE no hacer nada
      ENDIF
      Asignar el carácter a la variable alfanumérica
      transitoria
   ENDLOOP
   LET NOMBRE$ = variable alfanumérica transitoria
   END
3. (Hallar último espacio)
   BEGIN
   LOOP mientras los caracteres no explorados
   permanecen en NOMBRE$
      IF carácter = " "
         THEN anotar posición en una variable
         ELSE no hacer nada
      ENDIF
   ENDLOOP
   END
4. (Leer APELL$)
   BEGIN
   Asignar a APELL$ los caracteres a la derecha del último
   espacio de NOMBRE$
   END
5. (Leer NOMB$)
   BEGIN
   LOOP mientras caracteres no explorados permanecen en
   NOMBRE$ hasta último espacio
      SCAN caracteres
      IF el carácter no es una letra del alfabeto
         THEN no hacer nada
         ELSE asignar el carácter a NOMB$
      ENDIF
   ENDLOOP
   END
6. (Eliminar caracteres no alfabéticos de NOMB$)
   (Esto se ha manipulado arriba, en 5)
```

Este segundo nivel de refinamiento está ahora muy cerca de la etapa donde será codificado en un lenguaje de programación. Desarrollemos ahora 2 (es decir, convertir en mayúsculas todas las letras) en un tercer nivel de refinamiento y codifiquémoslo a continuación en lenguaje BASIC. Anteriormente ya habíamos encontrado un algoritmo que nos permitía realizar este proceso (véase p. 212).

3.er REFINAMIENTO

```
2. (Convertir en mayúsculas todas las letras)
   BEGIN
   READ NOMBRE$
   LOOP
   FOR L = 1 TO longitud de la variable
      READ carácter L
      IF carácter está en minúscula
         THEN restarle 32 al valor del
            carácter en ASCII
         ELSE no hacer nada
      ENDIF
      LET VARTRAN$ = VARTRAN$ + carácter
   ENDLOOP
   LET NOMBRE$ = VARTRAN$
   END
```

Este fragmento de programa en seudolenguaje ahora ya se parece bastante a un lenguaje de programación como para ser codificado. Nuestra versión de BASIC Microsoft no permite que los nombres de variables alfanuméricas sean palabras completas, de modo que en su lugar usaremos letras. Por lo tanto, NOMBRE$ se convierte en N$.

```
1000 REM SUBRUTINA DE CONVERSION A MAYUSCULAS
1010 INPUT "DE ENTRADA A NOMBRE";N$: REM SOLO
     PARA PROBAR
1020 LET P$ = " ": REM ASEGURAR QUE LA VARIABLE
     ESTA VACIA
1030 FOR L = 1 TO LEN(N$): REM INDICE DE BUCLE
1040 LET T$ = MID$(N$,L,1): REM EXTRAER
     CARACTER
1050 LET T = ASC(T$): REM HALLAR VALOR ASCII DE
     CARACTER
1060 IF T > = 97 THEN LET T = T - 32: REM
     CONVERSION A MAYUSCULAS
1070 LET T$ = CHR$(T)
1080 LET P$ = P$ + T$: REM P$ ES LA VARIABLE
     TRANSITORIA
1090 NEXT L: REM FIN BUCLE
1100 LET N$ = P$: REM AHORA N$ ESTA TODA EN
     MAYUSCULAS
2000 PRINT N$: REM SOLO PARA PROBAR
2010 END: REM SOLO PARA PROBAR. REEMPLAZAR
2020 REM 'RETURN' EN PROGRAMA VERDADERO
```

Este fragmento de programa se podría usar como una subrutina dentro de un programa principal. Lo hemos escrito para probar con una sentencia INPUT, otra PRINT, y varias REM y un END. No obstante, estas dos últimas habrían de eliminarse antes de que la subrutina se incorporara a un programa completo.

Complementos al BASIC



El Lynx posee una función, UPC$(A$), que convierte en mayúsculas las letras de A$, de manera que el programa se reduce así:

```
1010 INPUT "dé entrada a un nombre";N$
1020 LET N$ = UPC$(N$)
2000 PRINT N$
2010 REM "RETURN" AQUI EN EL
     PROGRAMA VERDADERO
```



El programa completo para el Spectrum es:

```
1010 INPUT "DE ENTRADA A UN
     NOMBRE";N$
1020 LET P$ = " "
1030 FOR L = 1 TO LEN N$
1040 LET T$ = N$(L)
1050 LET T = CODE T$
1060 IF T > = 97 THEN LET T =
     = T - 32
1070 LET T$ = CHR$ T
1080 LET P$ = P$ + T$
1090 NEXT L
2000 LET N$ = P$
2010 PRINT N$
2020 REM "RETURN" AQUI EN EL
     PROGRAMA VERDADERO
```

DRAGON 32

Este programa se puede ejecutar en el Dragon, pero los caracteres en minúsculas están reservados para la impresora, de modo que el problema no se plantea realmente



Reemplazar la línea 1010 por:

```
1010 INPUT "dé entrada a un nombre",
     N$
```

Ejercicios

■ Refinar todas las etapas anteriores hasta el punto en que se pudieran convertir en un programa en BASIC. El "seudolenguaje" que emplee no tiene por qué ser igual al nuestro, pero es una medida atinada continuar utilizando letras mayúsculas para los términos clave que probablemente corresponderán a palabras-sentencia en el lenguaje final (por ejemplo, LOOP, IF, LET, etc.). Emplee letras pequeñas para las operaciones que habrán de enunciarse más explícitamente cuando se codifiquen finalmente. Éstas las puede escribir en lenguaje corriente.

■ Luego de haber desarrollado los programas hasta un nivel de refinamiento satisfactorio, conviértalos en módulos de programa (subrutinas) en BASIC. Verifíquelos uno a uno empleando entradas ficticias e imprima sentencias que, si funcionan adecuadamente, se puedan eliminar luego.

Ejercicios de revisión

En estos ejercicios se aplican la mayor parte de las sentencias y funciones en BASIC que se utilizan más comúnmente. No hay ninguna pregunta capciosa ni se introduce ningún concepto nuevo. Si puede realizar todos estos ejercicios o la mayoría de ellos sin ninguna dificultad, puede usted considerar que está ya en camino de convertirse en un programador en BASIC hecho y derecho.

■ Escriba un programa para aceptar una entrada de dos números desde el teclado, para sumarlos e imprimir el resultado.

■ Asigne dos palabras (series de caracteres) a dos variables alfanuméricas y luego cree una tercera variable alfanumérica que concatene (es decir, que una entre sí) las dos palabras originales. Imprima a continuación la tercera variable.

■ Escriba un programa que le permita digitar cualquier palabra en el teclado y que después imprima la longitud de la variable en el mensaje LA PALABRA QUE HA DIGITADO TIENE * CARACTERES (* representa un número).

■ Escriba un programa que acepte un único carácter digitado en el teclado y que le diga luego cuál es el valor ASCII de ese carácter (en decimal).

■ Escriba un programa que le ofrezca el mensaje DIGITE UNA PALABRA y que después le responda con el mensaje LA ULTIMA LETRA DE LA PALABRA ERA * (* representa una letra).

■ Escriba un programa que le solicite que DIGITE EL NOMBRE Y PRIMER APELLIDO DE UNA PERSONA y que después le responda EL ESPACIO ERA EL *º CARACTER.

■ ¿Cómo modificaría el programa anterior para que imprimiera 3er en vez de 3º si el espacio estuviera en la tercera posición?

■ Escriba un programa que le solicite que DIGITE UNA ORACION y que después le responda con el mensaje LA ORACION QUE DIGITO TENIA * PALABRAS (dando por sentado que en la oración habrá una palabra más que el número de espacios).

■ Controle su ordenador para ver si se le han asignado caracteres (o gráficos especiales) a los valores de ASCII desde 128 hasta 255 (utilice un bucle y la función CHR$(X)).

La sala de espera

Los ordenadores transfieren la información a mayor velocidad que la que pueden manipular los dispositivos mecánicos. Ello se obvia con una memoria intermedia, el "buffer" o tampón

Los *buffers* (topes) que utilizan los trenes están diseñados para amortiguar el impacto, absorbiendo energía en muelles o pistones amortiguadores. Los ordenadores también poseen buffers y de alguna forma funcionan como aquellos topes, pues ayudan a "permanecer juntos" a los componentes del sistema del ordenador.

En el mundo de la informática este término se aplica con cierta vaguedad y se lo emplea en dos sentidos bastante diferentes. Para el programador, buffer significa una utilización especializada de memoria del ordenador, mientras que pàra el diseñador de circuitos significa una clase de amplificador de señales eléctricas. Los del segundo tipo, que denominaremos *buffers de señal*, son los de las tablas de conexiones.

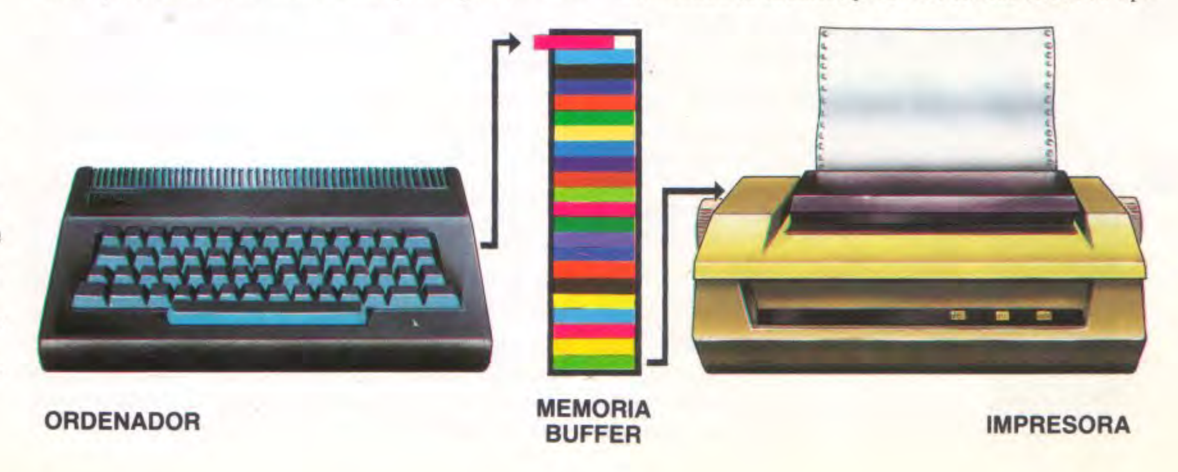
Buffers de memoria

Pensemos en un programa para tratamiento de textos que, entre otras cosas, puede desplazar un bloque de texto desde una parte de un "documento" en la memoria del ordenador hasta otra. El texto se compone de caracteres imprimibles y espacios, y de ciertos caracteres "no imprimibles", como el Carriage Return (retorno de carro). En la memoria del ordenador todos ellos están representados como códigos ASCII en binario. Para cada carácter se requiere un byte de memoria. Llevar los caracteres del bloque desde su antigua posición en la memoria hasta una nueva implica que se debe apartar otra parte de la memoria del ordenador para que sirva de zona de almacenamiento temporal de texto. Dicha zona de memoria apartada para una tarea específica se denomina *buffer*, o tampón.

Como segundo ejemplo, consideremos el problema de imprimir un documento creado con un procesador de textos. El documento podría constar de 15 000 caracteres separados, pero es evidente que no se le podrían enviar todos simultáneamente a la impresora para que los imprimiera: la velocidad de impresión de la mayoría de estos dispositivos no suele superar la cifra aproximada de 80 caracteres por segundo. Para orillar este inconveniente, se apartará parte de la memoria del ordenador, que cumplirá la función de *buffer de impresión*, bajo el control del software de tratamiento de textos. Este programa llenará primero este buffer con los caracteres a imprimir, y luego los irá enviando para su impresión a una velocidad adecuada para la impresora.

El buffer de impresión puede no ser muy grande, tal vez de una capacidad de sólo 128 a 256 bytes, pero, independientemente de sus dimensiones, los principios que lo rigen son los mismos. Primero se escribe en él un "bloque" de caracteres ASCII y después éstos se vuelven a enviar de uno en uno. El primer byte que se escriba en el buffer será, asimismo, el primer byte que se lea de él (como es lógico, deseamos que los caracteres se impriman en el mismo orden en que fueron digitados). Esta clase de buffer se conoce como buffer FIFO (*First In First Out*: primero en entrar, primero en salir). Una vez leídos todos los caracteres del buffer, el software lo llena con el siguiente bloque de caracteres destinados a la impresora.

Los buffers FIFO son muy empleados en la mayor parte de este tipo de software. Se utilizan cuando existe incompatibilidad de velocidades, no sólo entre ordenadores e impresoras, sino también entre ordenadores y unidades de disco flexible y entre ordenadores y teclados de ordenador. Aunque la asombrosa velocidad de procesamiento de los ordenadores significa que por lo general éstos pueden identificar las teclas digitadas en un tiempo menor que el empleado en pulsarlas, es posible que en algunas ocasiones el ordenador no pueda identificar las teclas y visualizar los caracteres correspondientes con la suficiente rapidez. Esto puede suceder si el ordenador está momentáneamente ocupado cumpliendo otra función (accediendo a un disco, por ejemplo). Cuando esto ocurre, lo común es incorporar en el sistema operativo del ordenador un *buffer de digitación adelantada*. Este buffer "recuerda" cuáles teclas se han pulsado y el ordenador las visualiza de inmediato. Lo más probable es que esta acción pase inadvertida al usuario, pero con ciertos sistemas ope-



Parada temporal
Uno de los usos más comunes del buffer es aquél entre el ordenador y una impresora, dado que ésta no puede dar salida a los caracteres a la misma velocidad en que el ordenador los envía. En consecuencia, los caracteres se almacenan en la memoria temporal hasta que este buffer esté completo, y entonces se le envía al ordenador una señal de "ocupado" para que éste deje de transmitir. Los contenidos de la memoria buffer se envían luego a la impresora por el mismo orden en que se recibieron, pero a una velocidad muy inferior. Cuando esta tarea ha terminado, el proceso vuelve a comenzar hasta que se haya impreso el texto completo



Si echamos agua con un cubo dentro de un tanque, del grifo de salida saldrá un chorro uniforme. Del mismo modo, el ordenador "vierte" datos en el buffer, el cual los libera a una velocidad muy inferior a la que funciona, por ejemplo, la impresora

rativos de disco, o con ciertos tipos de software aplicativo (que implican mucho procesamiento de información), podría producirse un ligero intervalo entre la pulsación de la tecla y su aparición en pantalla. Unos pocos sistemas operativos permiten encender o apagar el buffer de digitación adelantada, o incluso que el usuario altere las dimensiones del buffer.

La forma exacta en que está organizado el software para manipular los buffers varía según la función que hayan de cumplir los mismos, pero por lo general será necesario apartar unos pocos bytes para utilizarlos como contadores y banderas. Es indispensable saber cuántos bytes se han leído de un buffer lleno antes de escribir otros; de lo contrario, se podrían destruir datos importantes antes de que se los utilizara.

Buffers de memoria en hardware

Los lectores que posean impresoras pueden haber notado lo lentas que éstas parecen ser, especialmente al imprimir un listado de programa o un documento largo. La mayoría de los sistemas operativos de ordenador no pueden realizar ninguna otra labor mientras se está utilizando la impresora; por lo tanto, si la impresión requiere mucho tiempo, el usuario no tendrá otra opción que permanecer sentado frente a la pantalla a la espera de que la impresora finalice su cometido. Ahora muchos fabricantes ofrecen buffers de impresión accesorios, por lo general en forma de una caja que se conecta entre el ordenador y la impresora. En efecto, desde el punto de vista del ordenador, estas cajas consiguen que la impresora trabaje más aceleradamente. En realidad ésta no imprime con mayor rapidez, sino que esta mayor "velocidad" se debe a la memoria extra exclusiva (a veces hasta de 16 Kbytes), con su propio software incorporado, que poseen estas cajas. Cuando el ordenador ha de imprimir un archivo, le traspasa bytes a la impresora hasta que recibe una señal de "ocupado", que significa que ésta ya no puede aceptar ningún byte más. El ordenador, entonces, ha de esperar hasta que la línea "ocupada" se convierta en "falsa", lo que indicará que la impresora está nuevamente en condiciones de aceptar datos. Si bien por lo general las impresoras cuentan con un pequeño buffer de memoria incorporado, éste no suele poseer más de dos Kbytes y no permite que el ordenador envíe más datos hasta encontrarse vacío. Los accesorios de buffers de memoria en hardware contienen más memoria, de modo que pueden aceptar muchos más datos antes de enviarle al ordenador la señal de "ocupado". Si el buffer es lo suficientemente grande, puede tener capacidad para retener de una sola vez todos los datos a imprimir, con lo cual el ordenador podrá continuar con otras tareas mientras el buffer le envía los datos a la impresora a menor velocidad.

Con frecuencia la memoria se utiliza más bien como un gran soporte de archivo o bus para almacenar programas y datos, pero en lugar de ello se puede organizar en "pilas" o buffers. Las "pilas" son estructuras LIFO (*Last In First Out*: último en entrar, primero en salir), mientras que los buffers, como acabamos de ver, son estructuras FIFO (*First In First Out*: primero en entrar, primero en salir). Para las pilas se suele emplear la analogía del montón de platos apoyados sobre un resorte que existe a menudo en los mostradores de los self-services. Los platos están apilados en el rimero y el último que se coloque en él será el primero que se quite. Al igual que los buffers, las "pilas" también son zonas de memoria temporal y sólo se diferencian de los buffers en el orden en que se da entrada y se recupera la información. En los lenguajes de alto nivel (como en los intérpretes de BASIC, por ejemplo), las "pilas" se utilizan "internamente", pues se necesita almacenar temporalmente la información para ser cargada luego. Consideremos este fragmento de programa en BASIC:

```
FOR X = 1 TO 10
PRINT "X = ";X
FOR Y = 1 TO 10
GOSUB SCAN
NEXT Y
PRINT "C$ = ";C$
NEXT X
```

Éste es un ejemplo de bucles FOR...NEXT anidados. Cuando el intérprete de BASIC llega a la segunda sentencia FOR, necesita recordar cuál es la variable utilizada para el FOR anterior (X, en este caso) y por tanto "empuja" la información relativa al primer FOR dentro de una "pila". Cuando se ha completado el bucle interior, hace "saltar" la información desde el extremo superior de la pila y sabe que el FOR actual utiliza la variable X. Dado que los bucles FOR...NEXT se pueden anidar tan profundamente como sea necesario, podría necesitar empujar en la "pila" información para varios FOR. Cuando hace saltar la información de la "pila", obviamente necesita tener la información en orden inverso al orden en que fue empujada.

Por el contrario, los buffers organizan la memoria de modo que la primera información que entra es la primera información que sale. Los buffers se suelen utilizar para rutinas de entrada/salida y se emplean como "interfaces" entre rutinas o dispositivos que trabajan en unidades diferentes o a distintas velocidades. Por ejemplo, una rutina de entrada en BASIC podría trabajar en unidades de líneas, terminadas por un Carriage Return <CR>, pero el intérprete podría funcionar en las líneas por unidades de un carácter. Por lo general los buffers necesitan de un "señalador" que indique en qué lugar del buffer se ha de escribir el siguiente carácter. El señalador serían uno o varios bytes que contuvieran la dirección de aquel carácter. La dirección se incrementaría después de que se hubiera almacenado cada carácter.

Señales poderosas
La lógica interna del ordenador funciona a niveles TTL. Transistor-Transistor-Logic significa un 1 binario con cinco voltios y un 0 con cero voltios. No obstante, aunque dispositivos como la CPU pueden producir estos voltajes, no pueden generar corriente suficiente para activar todos los otros chips que podrían estar conectados a cada patilla. Por tanto, se conectan buffers de señal a las líneas de salida de la CPU para aumentar la cantidad de corriente. Los buffers de señal son pequeños chips, y cada uno de ellos actúa como un buffer para seis señales

Cálculo analógico

Los ordenadores analógicos, utilizados para controlar máquinas y procesos, reaccionan directamente a los cambios del mundo real, sin que sea preciso traducir la información a forma digital



Cortesía de Physical & Electronics Labs Ltd

Cuadro de mandos
Los ordenadores analógicos no utilizan lenguajes como el BASIC. Se "programan" empalmando diversos componentes eléctricos. Los componentes se fijan a la cara posterior de un tablero de circuitos. En la cara anterior se colocan los enchufes y conectores que permiten conectar entre sí los componentes seleccionados

Existen dos familias de ordenadores bastante diferentes, y hasta ahora sólo nos hemos estado ocupando de una de ellas: el ordenador digital, así llamado porque todas las instrucciones de un programa y todos los datos se representan utilizando "dígitos" binarios. La otra familia, la más antigua de las dos, es la de los ordenadores analógicos, que se programan de forma diferente: uniendo varios componentes electrónicos.

Topografía por analogía
Los planímetros como el de la ilustración, que data del siglo XVI, se utilizaban junto con una escuadra para efectuar la triangulación *in situ*. Anteriormente, los topógrafos tenían que tomar medidas y luego efectuar cálculos matemáticos para determinar la longitud del lado más lejano de su triángulo, pero gracias a este sencillo instrumento analógico su trabajo se volvió mucho más sencillo y exacto. Aún hoy en día se emplean instrumentos de nombre similar para medir la superficie de planos irregulares, como pueden ser las pieles de animales

El velocímetro constituye un ejemplo de este ordenador "analógico", especializado y sencillo; el nombre proviene del comportamiento "análogo" de la velocidad del coche respecto a la posición de la aguja en el dial, siendo aquélla directamente proporcional a ésta. Los ordenadores analógicos modernos pueden realizar muchas tareas y se basan en el tipo de componentes eléctricos que encontramos comúnmente en los aparatos domésticos: transistores, condensadores, resistencias e inductancias magnéticas. Cuando se comenzaba a desarrollar la electrónica, se descubrió que el comportamiento de los componentes eléctricos se parecía al de los dispositivos mecánicos. Por ejemplo, los ingenieros eléctricos descubrieron que las oscilaciones de corriente eléctrica que podían resultar al conectar entre sí una inductancia magnética y un condensador se asemejaban muchísimo a las oscilaciones de una pesa colgada de una cuerda. De hecho, la descripción matemática de ambos sistemas era idéntica: la base para el "cálculo analógico".

Algunos dispositivos analógicos se asemajan mucho a los sistemas que ejemplifican; por ejemplo, el modelo de un aeroplano que se emplea para un experimento de túnel de viento es una copia exacta, a escala reducida, de la armazón del avión. Otros modelos resultan muy distintos. Por ejemplo, un "modelo" de una situación de la vida real puede consistir tan sólo en una lista de fórmulas matemáticas, o puede ser un circuito eléctrico que reproduce el flujo de agua a través de una presa.

Las máquinas de calcular que utilizan principios analógicos demostraron por primera vez su importancia en 1630, cuando William Oughtred inventó la regla de cálculo. Los números estaban dispuestos sobre dos reglas de modo tal (estaban espaciados logarítmicamente) que el movimiento de una a lo largo de la otra equivalía a la multiplicación, y la respuesta simplemente se podía leer de la escala.

A fines del siglo XIX lord Kelvin diseñó un ingenioso dispositivo mecánico a manivela que se podía emplear para calcular las mareas altas y bajas de un puerto durante todo el año.

Luego, en 1930, se construyó en Estados Unidos una máquina electromecánica que podía resolver ecuaciones diferenciales generales (el tipo de ecuaciones que se descubre prácticamente en cualquier representación matemática del mundo real), en vez de la serie específica de ecuaciones diferenciales que Kelvin había conseguido resolver. La máquina la inventó Vannevar Bush, quien le dio la denominación de analizador diferencial.

Los primeros ordenadores analógicos

Los primeros ordenadores analógicos totalmente electrónicos entraron en funcionamiento en 1947, justo después de que nacieran los primeros ordenadores digitales. Hemos visto cómo los ordenadores digitales realizan cálculos aritméticos utilizando una combinación de puertas lógicas (véase p. 68). El ordenador analógico puede efectuar cálculos matemáticos simplemente utilizando la naturaleza de la electricidad. Si hay, por ejemplo, una corriente eléctrica de cinco amperios que fluye por un cable y una corriente de cuatro amperios que fluye por otro, y los dos cables están

unidos entre sí para converger en un único cable, la corriente del nuevo cable será de nueve amperios: la suma de las dos corrientes. De manera, por lo tanto, que el conjunto del comportamiento y las propiedades del flujo de corriente constituyen automáticamente un "sumador".

Hasta para las funciones matemáticas muy complicadas a menudo se hallan soluciones muy sencillas en circuitos elementales (un ejemplo es la "integración": hallar la superficie bajo una curva). Los ordenadores analógicos no se "programan" como las máquinas digitales; en cambio, se ha de construir un circuito que ejemplifique el programa a resolver. Todos los componentes disponibles se montan en la parte posterior de un "tablero", en cuya cara anterior están los enchufes y conectores para unir los componentes entre sí mediante cables. El aspecto de este "cuadro de mandos" recuerda mucho a un anticuado cuadro conmutador de teléfonos.

En un ordenador analógico se utilizan los diversos voltajes o corrientes para representar cantidades físicas como fuerza o velocidad, y los "valores" de los componentes eléctricos representan cosas como la masa de un coche o la fuerza de sus amortiguadores. Pero en un ordenador digital todos los datos se representan mediante series de impulsos, cinco voltios para el 1 binario y cero voltios para el 0 binario. Aquí hallamos una distinción más amplia entre los ordenadores analógicos y los digitales: en unos la información se puede almacenar de forma que satisfaga cantidades que cambian continuamente, pero en los otros los datos se almacenan en unidades "discretas" o individuales.

Algunos ordenadores personales (el BBC Micro, por ejemplo) poseen un conector para entrada analógica. En general, cuando los ordenadores digitales manipulan información analógica, como una temperatura o una fuerza, primero han de convertir los datos en forma digital.

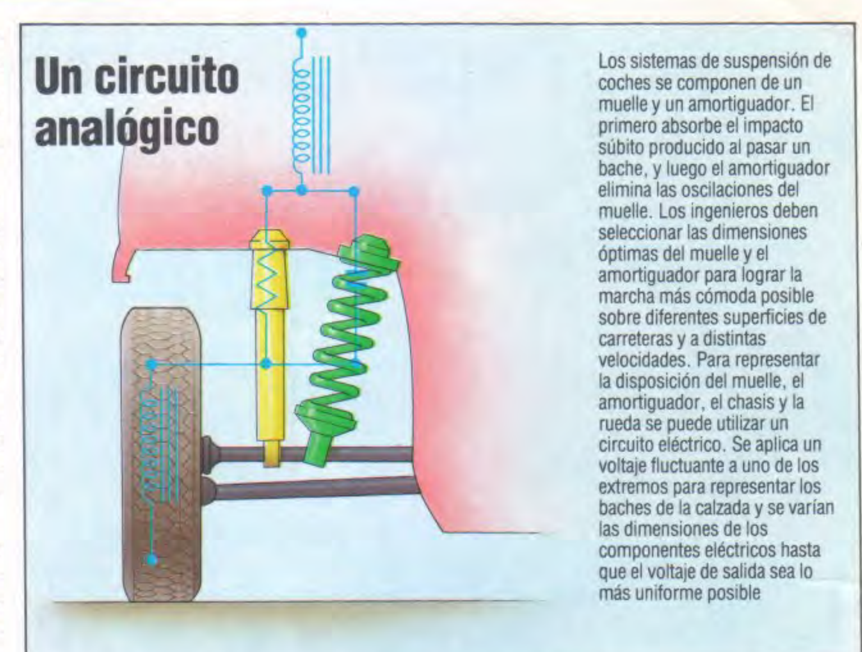
La gran ventaja de los sistemas digitales es que la información se puede procesar o transmitir sin pérdida de calidad. Si un impulso de cinco voltios se hace pasar a través de un circuito eléctrico, podría muy bien ser afectado por la distorsión inherente a todo circuito y salir, quizá, como 4,9 voltios. En un sistema analógico, en el cual las fluctuaciones de voltaje representan cambios en la información, esto podría significar, en la voz de un cantante, la diferencia que existe entre un la y un la bemol. Pero en un sistema digital, en el cual sólo hay dos señales posibles, cinco voltios o cero, cualquier señal próxima a cinco (por ejemplo, 4,9) se reconoce y se reproduce automáticamente como cinco voltios. De modo que los errores se corrigen y no se acumulan. Por el contrario, la acumulación de errores a medida que la señal pasa a través de circuitos sucesivos representa uno de los inconvenientes del ordenador analógico.

No obstante, este tipo de ordenadores saca provecho de representar las cantidades como valores variables de un voltaje o una corriente. Esto significa que una condición de entrada se puede cambiar rápidamente y que el sistema reflejará de inmediato las modificaciones consiguientes. No se necesita tiempo para codificar los datos en impulsos binarios, ni para procesarlos ni para, finalmente, volver a decodificarlos para producir la salida.

Esta característica es muy importante para las aplicaciones en las que resulta esencial producir respuestas rápidas. Por ejemplo, un piloto automático debe responder a una repentina corriente de viento durante un aterrizaje, cuando ya no hay tiempo para efectuar cálculos largos, ni siquiera a la velocidad a que funcionan los ordenadores digitales modernos. Los sensores detectan la corriente repentina, generando un voltaje de salida relativamente pequeño. El circuito del piloto automático responde de manera instantánea con un cambio del voltaje de salida relativamente grande, que de forma automática activa los alerones de las alas, de tal manera que el aparato consiga mantener la estabilidad sin mayores percances.

Los ordenadores analógicos se utilizan en muchas áreas del control industrial, en las cuales se han de manejar con sumo cuidado complicados sistemas mediante ajustes sutiles y continuos, como en una planta industrial química. Pero son menos conocidos que sus equivalentes digitales. Aunque algunas veces se emplean ordenadores analógicos sencillos en las escuelas, con fines pedagógicos, el tipo de aplicaciones para las cuales resultan idóneos revela que es muy poco probable que lleguemos alguna vez a ver un ordenador personal analógico. Los ordenadores analógicos se utilizarán siempre, pero serán los ordenadores digitales los que irán dominando el mercado a medida que vayan siendo más rápidos y más potentes.

Un circuito analógico



Los sistemas de suspensión de coches se componen de un muelle y un amortiguador. El primero absorbe el impacto súbito producido al pasar un bache, y luego el amortiguador elimina las oscilaciones del muelle. Los ingenieros deben seleccionar las dimensiones óptimas del muelle y el amortiguador para lograr la marcha más cómoda posible sobre diferentes superficies de carreteras y a distintas velocidades. Para representar la disposición del muelle, el amortiguador, el chasis y la rueda se puede utilizar un circuito eléctrico. Se aplica un voltaje fluctuante a uno de los extremos para representar los baches de la calzada y se varían las dimensiones de los componentes eléctricos hasta que el voltaje de salida sea lo más uniforme posible

Kevin Jones

Mensaje distorsionado



La manera en que se va alterando un mensaje cuando se transmite de oído a oído tiene gran similitud con los errores acumulativos de un circuito analógico

En un circuito digital sólo se puede pasar un número limitado de mensajes (concretamente 1 o 0). De modo que si se produjera cualquier distorsión, ésta se podría identificar fácilmente y eliminar en la siguiente etapa

Herman Hollerith

1860
Nace en Buffalo (estado de Nueva York)

1879
Se gradúa en la Universidad de Columbia y consigue empleo en la US National Census Office

1882
Se matricula en el Massachusetts Institute of Technology para realizar investigación

1883
Trabaja en Washington para la Patent Office

1884
Solicita sus primeras patentes para codificar información sobre una cinta continua de papel perforado

1887
Se adopta su sistema para procesar las estadísticas relativas a la mortalidad en Baltimore (Maryland)

1889
Se instala un sistema en la oficina de los jefes militares de sanidad para organizar las estadísticas médicas del ejército. Registra una patente por el invento de la tarjeta perforada individual

1890
Gana el concurso para provisión de equipos para procesar el censo de 1890. Obtiene el Doctorado en Filosofía por la Universidad de Columbia por su trabajo en el campo del procesamiento de la información

1900
Presenta una nueva generación de equipos perfeccionados

1901
Se utilizan los nuevos equipos para el censo agrícola

1905
Comienzan a caducar sus primeras patentes y la competencia empieza a luchar contra su monopolio

1911
Funda un holding: la Tabulating Recording Company

1914
El famoso administrador de empresas Thomas J. Watson asume la dirección del holding

1924
La empresa cambia de nombre y pasa a denominarse International Business Machines (IBM)

1929
Muere en Washington, D.C.


Cortesía de IBM Uk Ltd

El inventor que colocó las cifras de la población de Estados Unidos en tarjetas perforadas y fundó la mayor empresa de informática del mundo

Hollerith nació en Estados Unidos en 1860. Después de graduarse en la Universidad de Columbia consiguió empleo en la US National Census Office (Oficina Nacional de Empadronamiento), ayudando a elaborar estadísticas tomadas del censo de 1880. Todo el trabajo se realizaba a mano y resultaba concienzudamente lento; lo era tanto que, de hecho, cuando diez años más tarde llegó el momento de realizar el censo siguiente, en la oficina todavía no habían terminado de tabular los resultados del anterior. Hollerith sabía que su mayor aptitud era su poder de invención; con el objeto de entrenarse como inventor y desarrollar sus cualidades creativas, abandonó su trabajo en la National Census Office e ingresó en la Patent Office (Oficina de Patentes) de Washington.

La primera idea de Hollerith fue la de codificar información en cinta de papel. La cinta de papel estaba dividida en "campos" mediante marcas de tinta. Cada campo representaba diferentes categorías: por ejemplo, varón o mujer o blanco o negro. La presencia de un agujero en el campo varón/mujer significaba que la persona era un varón, mientras que su ausencia implicaba que se trataba de una mujer, y así sucesivamente. Estos agujeros después se podían "leer" mediante una máquina. Sus primeras patentes se produjeron en 1884 y dedicó los años siguientes a perfeccionar el sistema. Empezó por procesar la información relativa a las estadísticas de sanidad de las ciudades norteamericanas, que se estaban desarrollando rápidamente, y de la administración militar.

Cinco años después, en 1889, perfeccionó la idea de la cinta de papel perforado utilizando tarjetas separadas para cada persona. Las tarjetas eran del tamaño de los billetes de un dólar; se dice que esto se debió, en parte, a que los únicos equipos que se pudieron adaptar habían sido construidos para manipular dinero. Originalmente los agujeros eran redondos y se hacían con el punzón que utilizaban los conductores de autobuses para perforar los billetes, pero luego se construyeron punzones especiales para cortar un agujero cuadrado de 6 mm. Así se podía incluir gran cantidad de información en una sola tarjeta.

La ventaja que ofrecen las tarjetas individuales sobre la cinta continua es que, además de obtener los totales, la información se puede clasificar. Por ejemplo, a usted tal vez le interese saber qué cantidad de mujeres blancas de 80 años de edad viven en la ciudad de Nueva York. Se podrían clasificar todas las tarjetas y las que tuvieran agujeros perforados en estos tres campos se podrían separar mecánicamente del resto. Estas primeras máquinas sólo podían producir totales, pero posteriormente Hollerith introdujo la suma y otras operaciones aritméticas sencillas.

El éxito comercial llegó en 1889, cuando el Bureau of Censuses (Departamento de Censos) convocó un concurso para proveerse de un sistema de equipos que procesara el censo que se realizaría al año siguiente. Los sistemas se probaron volviendo a tabular los datos obtenidos del censo anterior. La convocatoria la ganó el equipo de Hollerith. Para entonces todas sus máquinas estaban amparadas bajo patentes y él aprovechó su monopolio para cargarle al gobierno 65 céntimos por cada mil tarjetas procesadas. Aunque cada habitante de Estados Unidos tenía su propia tarjeta individual, Hollerith sólo tardó dos años en hacer el trabajo. Anunció que la población del país era de 56 millones de habitantes y presentó la factura al gobierno.

Cuando llegó el momento de realizar el censo de 1900, Hollerith había desarrollado una maquinaria muchísimo más eficaz, pero se negó a rebajar su tarifa. Cuando caducaron sus patentes, el gobierno buscó otras empresas, pero Hollerith superó a la competencia fundando su propia compañía, que posteriormente se convertiría en la IBM (International Business Machines), firma que hoy está a la cabeza de las empresas fabricantes de ordenadores en todo el mundo.

Juegos de tarjetas
El método original de Hollerith para representar la información se sigue utilizando en la actualidad, un siglo después, si bien se ha modificado el formato de la tarjeta. Las modernas tarjetas perforadas constan de 12 filas y 80 columnas. Las máquinas tabuladoras utilizaban el sistema decimal y, por tanto, cada ficha podía almacenar 80 números. Los caracteres alfabéticos se crearon con la "multiperforación": efectuando más de un agujero en una columna. Los ordenadores también aceptan fichas perforadas en sistema binario

